APEX Profiling Float

User Manual

P/N 301308, Rev. 11

Teledyne Webb Research,
a Business Unit of Teledyne Instruments, Inc.
49 Edgerton Drive
North Falmouth, MA 02556
U.S.A.
Tel: (508) 563-1000

http://www.teledynemarine.com/webb-research/

MANUAL CHANGE LOG

The following are the principal changes to this manual from that of Revision 10 to document updates to the Teledyne Webb Research APEX profiling float firmware as of 2.14.6.

- Updates made to configuration parameter descriptions, syntax, range, and default, as applicable for Idle mode, Mission mode, Recovery mode and Emergency mode.
- **TelemetryDays** and **TelemetryInterval** configuration parameters and their descriptions, syntax, range, and default added to the Surface phase of Mission mode.
- Multi-mission science logs that transmit a cross section of the science logs ranging from the newest to the oldest where the float hops through the list of files added.
- RBR*argo*^{3} C.T.D added as an available CTD.
- Description and example for Down Time as specified by the **DownTime** configuration parameter added as applicable to the Park Descent, Park and Deep Descent phases of Mission mode.
- Description and example for Up Time as specified by the **UpTime** configuration parameter added as applicable to the Ascent and Surface phases of Mission mode.
- **IceDescentCycles**, **IceDescentPressure**, **IceDescentCount**, **IceDescentTimerInterval**, **IceDescentNudge**, **IceDescentTimeout**, and I**ceAscentTimeout** configuration parameters and their descriptions, syntax, range, and default added for ice avoidance.
- Sensors added to list for APEX-BGC Biogeochemical profiling float.
- Update made to procedure for starting a mission from a connected PC.
- Setting up communications moved to Appendix F.
- Seascan RAFOS and RTC sensors added.
- TriOS RAMSES G2 Radiometer sensor added.
- Mission Sequencing configuration added.

QUICK-START GUIDE

This quick-start guide encompasses the steps required to deploy the Teledyne Webb Research APEX profiling float right out of its crate without having to read this entire manual. Perform the steps below, and if desired, refer to the referenced page numbers of the manual for more detail. When viewing this manual as a PDF using Adobe Reader, you can click the step numbers or the referenced page numbers to go to the detailed descriptions.

NOTE *The APEX profiling float is shipped ready for deployment without any operator intervention. However, Teledyne Webb Research requires that a float self test be performed prior to deployment to maintain warranty and to verify that the float is operational after shipment to the deployment location as described in "Pre-deployment Testing" on page 2-12. In the event this self test is not performed prior to deployment and the results provided to Teledyne Webb Research, the warranty may be void. The Reset tool can be used to start the mission and verify that the self test has passed. Similarly, connecting a PC to the float to start the mission allows you to verify that the self test has passed before deploying the float plus view the test results. It is not necessary to use the Reset tool or a connected PC to start the mission. The self test will be performed and the mission will start if the float is deployed directly from its crate. To start the mission using the Reset tool, refer to "Starting the Mission using the Reset Tool" on page 2-5. To start the mission using a connected PC, first refer to "Connecting a PC to the Float" on page 2-10 and connect a PC to the float. Then refer to "Starting the Mission with a Command from a Connected PC" on page 2-6 and start the mission. After starting the mission using the reset tool or a connected PC, the float should be deployed after the self test is complete but before the end of the Mission Prelude phase which by default is 120 minutes, otherwise unexpected results may occur. For information on the Mission Prelude phase, refer to SECTION 3: "Operating Modes."*

STEP 1: **Unpack the float from its crate and allow it to warm up indoors if it has been stored at a temperature of -2°C or less. Refer to page 2-1.**

STEP 2: **Remove the plastic bag from the CTD, and then remove any protective caps and plugs from the optional sensors. Save these items to use again if the float is to be stored at a later time. Refer to page 2-3.**

STEP 3: **Verify that the air compartment of the bladder is *deflated*, and then deploy the float. Refer to page 2-8.**

Table of Contents

List of Figures

List of Tables

Battery Hazard Warnings

Teledyne Webb Research APEX profiling floats use either lithium or alkaline batteries as a power source. Since the use of either of these battery types can result in a fire, explosion or other hazards, only properly trained personnel should select and install the batteries in an APEX profiling float. Because there is no way to completely eliminate these hazards, Teledyne Webb Research shall not be liable for any consequential, special, incidental, indirect, multiple, administrative, or punitive damages, or any damage of an indirect or consequential nature arising out of, or related to, the use of lithium or alkaline batteries in performance of the Teledyne Webb Research APEX profiling floats.

Alkaline Battery Hazard

When alkaline batteries are confined in a sealed housing, there is a small chance that a combustible gas can accumulate and cause a fire or explosion. To minimize the chances of this occurrence, Teledyne Webb Research has added a catalyst inside the APEX profiling float housings which recombines hydrogen and oxygen into water. The housings are also designed to relieve excessive internal pressure buildup by having the upper end cap vent.

Lithium Battery Hazard

Lithium batteries pose an even greater risk than alkaline batteries, as contact with water can result in a fire or explosion. Although APEX profiling floats are completely sealed, there is always a small chance that a leak has occurred.

Maximum Depth Warning

Deploying an APEX profiling float to a depth that exceeds its 2000-meter depth rating will result in a catastrophic failure. Do not exceed the depth rating of an APEX profiling float.

Communications Loss Warning

Communications with an APEX profiling float is over an Iridium satellite network. RUDICS and dialup modems require a primary and an alternate phone number as well as a username and password combination. This information is stored on the APEX profiling float and should be verified through communications testing before deploying it, as unknown or incorrectly stored values can prevent communication, resulting in its loss.

Preface

The APEX (Autonomous Profiling Explorer) profiling float is an autonomous drifting profiler that acquires water column profile data, such as water conductivity, temperature and pressure, while ascending from a depth of up to 2000 meters. The APEX profiling float also includes the hardware and firmware to support a high level of operator flexibility and multiple functions.

Purpose of this Manual

This manual provides information on the setup, testing and deployment of an APEX profiling float. It encompasses its latest operational features as of the current revision. Updates to this manual can be obtained by request at apexsupport@teledyne.com as features are continually added or updated.

This manual is divided into the following seven sections and six appendices:

Section 1: Overview. Provides an overall description of the APEX profiling float, including the APEX-BGC Biogeochemical and APEX-AMS Advanced Multisensor profiling floats.

Section 2: Setup, Test and Deployment. Provides instructions on how to unpack, prepare for deployment, activate, test, and deploy the APEX profiling float.

Section 3: Operating Modes. Illustrates and describes the operating modes performed by the APEX profiling float.

Section 4: Files. Describes the configuration and log files included on the APEX profiling float.

Section 5: Modifying the Mission Plan. Describes how to modify the mission plan locally before deploying the float and remotely after deploying the float. In addition, it lists and describes the available commands that can be used to control some of the float functions and access its files.

Section 6: CTDs. Provides information on the available CTDs, including their hardware configuration and how to read their data log files and specify their sampling behavior.

Section 7: General Maintenance. Provides some cleaning and battery replacement recommendations for the APEX profiling float.

Appendix A: Optional Sensors. Provides information on the available optional sensors that can be installed on the float, including manufacturer references and sensor configuration parameters.

Appendix B: Ice Avoidance. Illustrates and describes the ice avoidance function that can be enabled before or after deployment of an APEX profiling float.

Appendix C: Time of Day Operation. Illustrates and describes the time of day operation.

Appendix D: Argos Telemetry. Provides information on the optionally available Argos telemetry link, including descriptions of the transmitted test and data messages. In addition, instructions are provided on how to convert Argos message files to text files.

Appendix E: Hyper Retract/N2 Compensation. Describes the hyper retract feature which can be used with or without N2 compensation hardware on a standard APEX profiling float.

Appendix F: Setting up Communications. Provides information on how to set up RUDICS/PSTN and SBD communications with the float.

Appendix G: APEX-EM Electromagnetic Float. Continually acquires water current, temperature and salinity profiles.

Appendix H: Diagnostic Messages Example. Provides an example of the progress and diagnostic messages displayed when starting the mission from a connected PC.

Warnings, Cautions and Notes

Where applicable, warnings, cautions and notes are provided in this manual as follows:

WARNING *Identifies a potential hazard that could cause personal injury or death to yourself or to others.*

CAUTION *Identifies a potential hazard that could be damaging to equipment or could result in the loss of data.*

NOTE *Recommendations or general information that is particular to the material being presented or a referral to another part of this manual or to another manual.*

Documentation Changes

Teledyne Webb Research reserves the right to make changes to the design or specifications of the APEX profiling float at any time without incurring any obligation to modify previously delivered units. In addition, while considerable effort has been made to ensure that the information in this manual is accurate and complete, Teledyne Webb Research assumes no liability for any errors or omissions.

Returns

Before returning an APEX profiling float for any reason, contact Teledyne Webb Research customer service for a Return Material Authorization (RMA) and return instructions. Additional information can be found at http://www.teledynemarine.com/SitePages/WebbResearchRMA.aspx.

WARNING *Do not attempt to open the APEX profiling float housing before returning the float to Teledyne Webb Research. If the float was recovered from the ocean, it may contain water. This situation presents a safety hazard due to the possible chemical reaction of the batteries with the water. Before returning a float to Teledyne Webb Research, contact Teledyne Webb Research customer service for instructions on how to properly prepare the float for shipment.*

Customer Service

We welcome your comments and suggestions for improving our products and documentation as well as developing better ways of serving you. Should you require service or support for an APEX profiling float, contact Teledyne Webb Research customer service using any of the following means:

Site URL: http://www.teledynemarine.com/apex-argo

Mail: 49 Edgerton Drive
North Falmouth, MA 02556

Telephone: (508) 563-1000

SECTION 1: Overview

The Teledyne Webb Research APEX (Autonomous Profiling Explorer) profiling float, or "float" for short, is an autonomous drifting profiler that acquires water column profile data, such as water conductivity, temperature and pressure, while ascending from a depth of up to 2000 meters. When at the surface, the collected data are transmitted over the Iridium satellite network to a shore based server. Depending on the type of batteries used, the sensors installed and the sensor sample rates, a float typically has a life ranging from four to six years during which it can descend to its maximum depth and ascend to the surface approximately 150 to 250 times.

1.1 Operation

The APEX profiling float automatically descends and activates when it is deployed, performs a self test, which includes an ascent to the surface, and starts a mission by descending to its "Park" depth, which is usually 1000 meters as shown in Figure 1-1. It will drift for typically 10 days at the Park depth and then descend to a deeper depth, the "Profile" depth, which is usually its maximum rated depth of 2000 meters. After reaching the Profile depth, the float will ascend to the surface while acquiring profile data. The trip

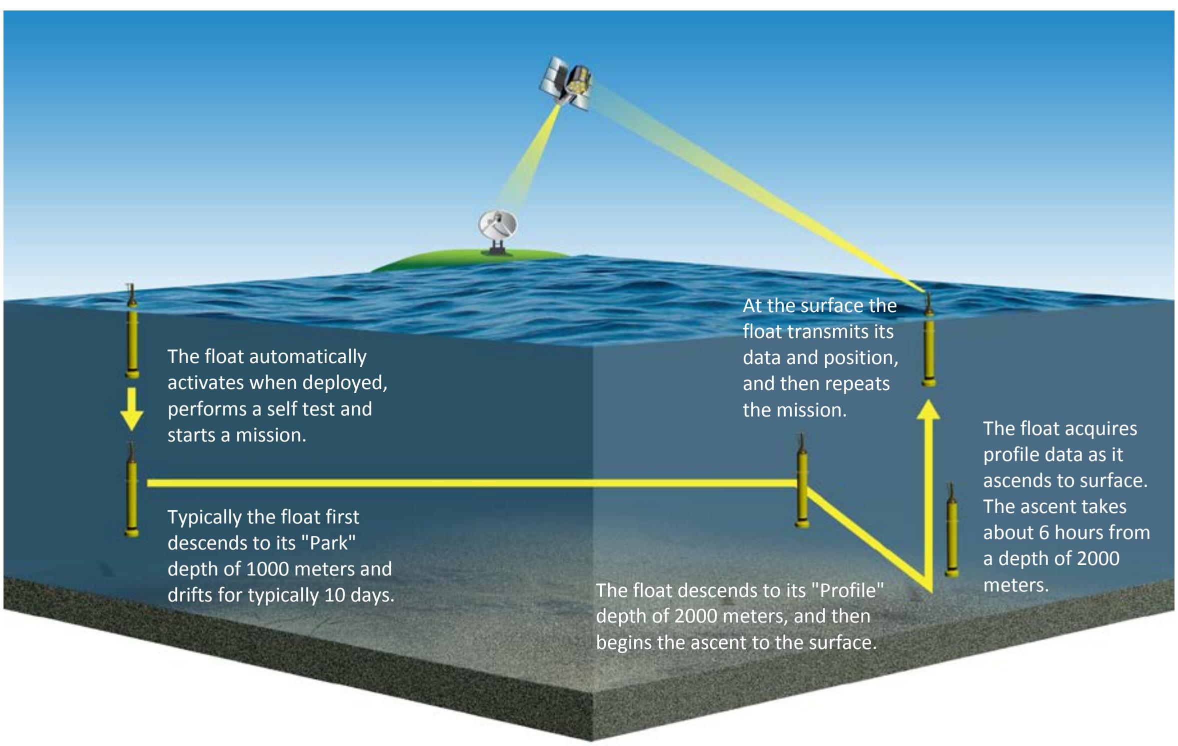


Figure 1-1: APEX Profiling Float Profiling Cycle

to the surface from 2000 meters takes approximately 6 hours. When at the surface, a GPS position fix is obtained, and the data and position of the float are transmitted over the Iridium satellite network to a shore based server. While at the surface, the float can accept user commands or download configuration files over the Iridium satellite network to modify its mission configuration parameters which include the Park and Profile depths, the ascent rate and other parameters. In the same manner the float can modify its sample configuration parameters which specify its sampling behavior. After transmitting the information, the float repeats the mission beginning with its descent to the Park depth.

1.2 General Description

An APEX profiling float is shown in two configurations in Figure 1-2, one with a "pumped" type CTD, and the other with a "non-pumped" type CTD. Otherwise the floats are similar. The float is composed primarily of a housing, oil and air bladders, the CTD, and a helical antenna. It is also capable of interfacing with additional, optional sensors.

NOTE *For information on APEX profiling floats with Argos telemetry, refer to APPENDIX D: "Argos Telemetry."*

1.2.1 Housing

The housing is constructed of anodized aluminum and coated with yellow epoxy paint. Carbon fiber housings are also available. It includes a damper disk which minimizes bobbing motion at the surface and has a hole to pass a line through during deployment. Contained inside the housing are all the electronics, hydraulics, battery packs, and mechanical systems required for operation. The float housing is rated to 2000 meters of water depth; however, lighter, thinner walled housings are available for shallower applications, and because they are lighter, they can include a larger battery payload. Upper and lower end caps seal the housing. In the unlikely event that the internal pressure increases excessively, as from battery gassing, either end cap will vent the excess pressure by temporarily breaching. A zinc anode is attached to the upper end cap for corrosion protection. The upper end cap also includes a seal plug which allows manual venting of the housing. Venting should *always* be performed before opening the housing.

1.2.2 Bladder

A rubber bladder that is composed of two compartments, one for oil and one for air, is contained inside a cowling which is mounted to the lower end cap of the float. The oil compartment fills with oil from a reservoir inside the housing; the air compartment fills

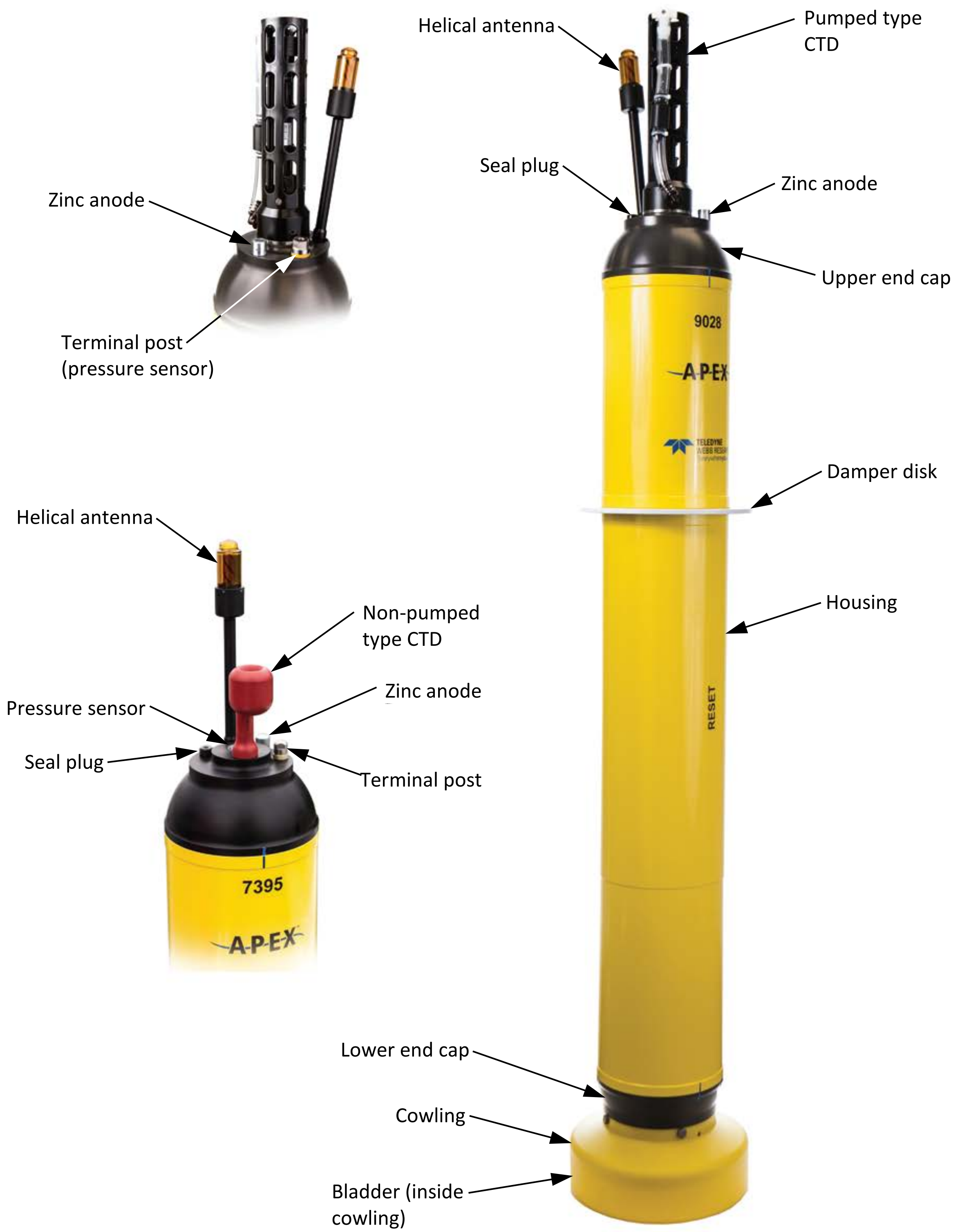


Figure 1-2: The APEX Profiling Float—Shown with a Pumped Type CTD and with a Non-pumped Type CTD

with air from inside the housing. Filling the oil compartment causes the float to ascend from depth, as its volume increases without changing its weight; removing the oil causes it to descend, as the volume decreases. This method works because increasing the volume of the float without changing its weight decreases its density, and decreasing the volume increases its density. The ascent rate is controlled by adjusting the amount of oil in the oil compartment. The air compartment is filled only when the float is near the surface to ensure that the antenna remains above water.

1.2.3 CTD

The CTD, including its pressure sensor, is mounted to the upper end cap. Either the pumped type CTD or the non-pumped type CTD can be installed.

For the pumped type CTD, a pump provides water flow to the temperature and conductivity sensors through a U-shaped flow path during every measurement cycle. The U-shaped path prevents flow due to water currents when the pump is not running, and ensures consistently compensated measurements of temperature and conductivity to yield accurate salinity calculations. It also includes anti-fouling devices. The pump is off between measurement cycles to conserve power. It is also off when the float is near the surface to prevent surface contaminants from being ingested. It remains off while the float is at the surface. The pressure sensor is a premium strain gauge type, and temperature is measured using a thermistor.

The non-pumped type CTD uses an inductive type conductivity sensor that is resistant to contaminants and therefore can measure conductivity all the way to the surface. The pressure sensor is a piezo-resistive transducer with a nickel based super alloy diaphragm that resists corrosion, and temperature is measured using a thermistor that is co-located with the conductivity sensor.

The CTD can also include integrated sensors. For example, it can include sensors to measure pH or dissolved oxygen.

1.2.4 Helical Antenna

The helical antenna is mounted to the upper end cap. This antenna connects both to a GPS receiver for float position and to an Iridium modem for data communications. A strobe light can be optionally included with the antenna.

1.3 General Operating Theory

The major functional components of the float are a Controller board which controls its operation, an RF board for communications, a GPS receiver which acquires float position, an Iridium modem which enables communications with the float over the Iridium satellite network, and alkaline or lithium battery packs which power the float. An air pump is used to inflate the air compartment of the bladder. Opening a valve deflates it. A buoyancy pump is used to push oil into and pull oil out of the oil compartment.

1.3.1 Controller Board

The operation of the float is controlled by the Controller board. This board controls the air and buoyancy pumps and interfaces with the GPS receiver, the Iridium modem, the CTD, and any additionally installed science sensors. It also includes the float's microcomputer, an internal clock, operational firmware and program and data memory, and it inputs and processes signals from a leak detector and internal temperature, pressure and humidity sensors. In addition, an independent hardware watchdog timer is included which is automatically serviced every 5400 seconds to prevent rebooting of the firmware. This function enables user programmed timer intervals to extend past this period without rebooting.

1.3.2 RF Board

The RF board serves as a platform for the GPS receiver, the Iridium modem and other communications devices if required.

1.3.3 GPS Receiver

The GPS receiver acquires float position and performs time synchronization of the internal clock on the Controller board with the GPS satellite time if needed when the float is on the surface. The receiver is a Garmin GPS 15xL GPS Receiver which shares the antenna with the Iridium modem and connects to the RF board.

1.3.4 Iridium Modem

The Iridium modem provides two-way data communications with a shore based server over the Iridium satellite network. The RUDICS/PSTN modem is an Iridium 9523 Satellite Transceiver which shares the antenna with the GPS receiver and connects to the RF board. The SBD modem is a 9602 Satellite Transceiver.

1.3.5 Battery Packs

Alkaline or lithium battery packs, or both, are used to power the float, depending on the operator requirements. Refer to "Battery Hazard Warnings" on page xvii for information pertaining to the existing hazards for these battery types.

1.3.6 Air Pump

The air compartment of the bladder is inflated by using the air pump to pump air into it from the housing, and it is deflated by opening a valve to allow the air back into the housing.

1.3.7 Buoyancy Pump

The oil compartment of the bladder is filled by using a piston to push oil into the compartment from an oil reservoir and emptied by using the piston to pull oil out of the compartment back into the reservoir. The piston is a component of a buoyancy pump that is powered from a gear motor. The buoyancy position is tracked using counts, typically from 147 to 3720. At 147 the piston is pulled all the way in which empties the bladder, increases the density of the float and causes it to descend. At 3720 the piston is pushed all the way out which fills the bladder, decreases the density of the float and causes it to ascend.

1.4 Other Float Configurations and Options

The standard configuration for the APEX profiling float includes a 2000-meter depth rated housing, a CTD, an Iridium modem, and a GPS receiver. However, other configurations can be built that include one or more of the following:

- Light weight, noncorrosive carbon fiber housing.
- Iridium Short Burst Data (SBD) low latency, short messaging two-way satellite communications link.
- Optional sensors, including an oxygen sensor, a transmissometer, a fluorometer, and others.
- Strobe light.
- Argos telemetry.

NOTE *For information on APEX profiling floats with Argos telemetry, refer to APPENDIX D: "Argos Telemetry."*

1.5 Additional Float Types

Four additional float types are available to suit a variety of specific research requirements:

- APEX Deep
- APEX-BGC Biogeochemical
- APEX-AMS Advanced Multisensor
- APEX EM Electromagnetic
- Customer specials

1.5.1 APEX Deep Profiling Float

The APEX Deep profiling float is rated to 6000 meters and is contained in a 17-inch diameter glass sphere housing encapsulated in a yellow ribbed hard hat.

1.5.2 APEX-BGC Biogeochemical Profiling Float

The APEX-BGC Biogeochemical profiling float is an APEX profiling float that can include any one or more of the following sensors:

- Oxygen sensor
- Fluorometer
- C-Rover Transmissometer
- Radiance radiometer
- Irradiance radiometer
- Nitrate sensor
- Compass
- pH

The oxygen sensor measures dissolved oxygen, and the fluorometer measures chlorophyll fluorescence and particulate backscattering. A connector on the outside of the housing is included with the float for connecting the cable from the fluorometer. The oxygen sensor is internally connected. The radiance and irradiance radiometers both measure spectral light, the C-Rover transmissometer measures optical transmittance of sea water, and the compass provides heading and tilt measurements. In addition, the pumped type CTD can optionally include a pH sensor which measures pH in salt water.

For more information on these sensors, including where they are mounted on the float, refer to APPENDIX A: "Optional Sensors."

1.5.3 APEX-AMS Advanced Multisensor Profiling Float

The APEX-AMS Advanced Multisensor profiling float is an APEX profiling float that may include other optional sensors, such as the RAFOS Hydrophone. For information on the RAFOS Hydrophone, including where it is mounted on the float, refer to APPENDIX A: "Optional Sensors."

1.5.4 APEX-EM Electromagnetic Profiling Float

The APEX-EM Electromagnetic profiling float acquires water column current velocity data along with other profile data, such as water conductivity, temperature and pressure.

1.5.5 Customer Specials

For specific research requirements that require one or more sensors that are not currently available with an APEX-BGC Biogeochemical or APEX-AMS Advanced Multisensor profiling float, a customer special of an APEX profiling float can be provided that integrates these sensors. The float can be customized with the required hardware, firmware and internal and external connections.

1.6 Deployment Methods

APEX profiling floats are deployed from research vessels as well as merchant ships, such as Volunteer Observing Ships (VOS) or Ships of Opportunity (SOOP). As merchant ships are continuously under way at speeds of up to 25 knots or more and have decks as high as 25 meters above the waterline, floats can be packaged in biodegradable cardboard containers which can be hand deployed under these conditions using lines to lower them over the side. A float can also be deployed from a stationary research vessel by passing a line through the hole in the damper disk and carefully lowering the float over the side while holding both ends of the line. Once in the water, one end of the line is released and pulled through and out of the hole. A line should never remain attached to a float. A third deployment method is by air. For instructions on how to air deploy a float, contact Teledyne Webb Research customer service.

SECTION 2: Setup, Test and Deployment

This section provides instructions on how to unpack, prepare for deployment, activate, test, and deploy the Teledyne Webb APEX profiling float.

2.1 Unpacking and Inspection

Included with each shipment of one or more floats is an accessories set as shown in Figure 2-1. The set includes the following items:

Reset tool. The Reset tool is used to manually start the mission beginning with a self test.

Cowling plug. The cowling plug is used to seal the large hole at the bottom of the cowling. It is a spare component.

Small cowling plugs. The small cowling plugs are used to seal the small holes at the bottom of the cowling. They are spare components.

RS-232 Current Loop Converter. The RS-232 Current Loop Converter enables bidirectional communications between the float and a PC. It includes a 25-pin to 9-pin serial port adapter for a PC that uses a 9-pin RS-232 serial port.

15 VDC Power Supply. The 15 VDC Power Supply is used to power the RS-232 Current Loop Converter.

Communications cable. The communications cable is used to connect the RS-232 Current Loop Converter to the float.

Universal AC adapter plugs. Four universal AC adapter plugs enable connection of the 15 VDC Power supply to most any type of power outlet.

The floats are carefully packed in wooden crates. The accessory items are packed in a plastic bag and included with one of the floats. Every effort is made to pack the floats to protect them during shipment. However, carefully inspect the crates for any signs of external damage. After inspecting the crates, locate and remove the packing list and verify that all of the items are included. Open the crates and inspect each float for damage that might have occurred during shipment. If any damage is found, either to a crate or to a float, immediately report the damage to Teledyne Webb Research and to the freight carrier. Floats must also be tested to maintain warranty as described in "Running the Self Test from a Connected PC" on page 2-7.

Figure 2-1: APEX Profiling Float Accessories

2.2 Preparing the Float for Deployment

To prepare the float for deployment:

1. Remove the float from its crate, and using the foam cradles packed with the float, lay it on a flat, horizontal surface.
2. Allow the float to warm up indoors if it has been stored at a temperature of -2°C or less.
3. Remove any plastic bags covering the CTD as shown in Figure 2-2.

Figure 2-2: Plastic Bag to be Removed from CTD

4. For a float with a pumped type CTD only, remove the cap and two plugs from the CTD along with any protective caps from the optional sensors as shown in Figure 2-3. Save the caps and plugs to use if the float is to be stored at a later time.

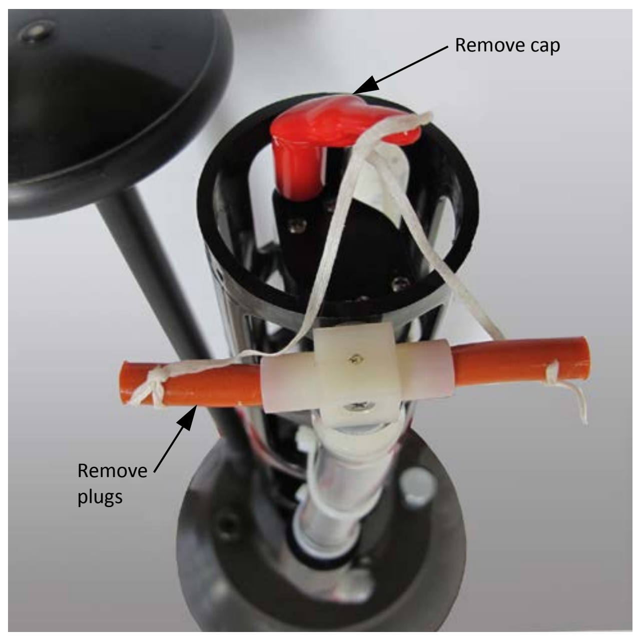


Figure 2-3: Cap and Plugs to be Removed from Pumped Type CTD

2.3 Starting the Mission

The float will start a mission whenever any one of the following operations is performed:

- The float is deployed directly from its shipping container and pressure activates.
- The Reset tool is swiped against the float housing.
- A command to start the mission is sent from a connected PC.

Once the mission starts, the float will enter the Mission Prelude phase. During the Mission Prelude phase, the float will run an internal self test as **PreludeTime** allows, obtain its GPS position, perform time synchronization of the internal clock with the GPS satellite time if needed, upload log files to the server, receive configuration files from the server, and wait for **PreludeTime** to expire. For information on the Mission Prelude phase, **PreludeTime** and all of the other configuration parameters, refer to SECTION 3: “Operating Modes.”

When pressure activation is used to start a mission *after* deployment, the float will start the internal self test at the activation pressure while it ascends to the surface. The GPS and Telemetry components of the self test are performed only if the float can establish satellite communications within **PreludeTime**.

When the Reset tool or a command from a connected PC is used to start the mission *before* deployment, several events can be monitored: the float will vibrate as the buoyancy pump runs, the compartments of the bladders in the cowling will fill with oil and air, and log files will appear on the server. Once any of these events is observed, the float is ready for deployment. When starting the mission using the Reset tool or a command, the float *must* be outside with a clear view of the sky. It should also be deployed within 120 minutes after starting the mission as described in “Deploying the Float” on page 2-8.

NOTE *The float will perform a self test and some mission preparation tasks for each of these operations prior to the actual mission start. The self test can also be run from a connected PC as described in “Running the Self Test from a Connected PC” on page 2-7. This option does not start the mission and is recommended to verify adequate battery capacity when a float has been in storage for an extended period of a year or more.*

CAUTION *When the self test is running, both the oil and air compartments of the bladder will fill causing the bottom of cowling to expand slightly. If the float is resting on deck in its upright position, it could topple over causing damage to the CTD or to the antenna. Before starting the mission with the Reset tool or the PC, or before running the self test from a connected PC, secure the float such that it will remain stable during the self test.*

The self test performs numerous checks, several of which are the following:

- CTD current draw
- RF board current draw
- Humidity
- Leak detection
- Internal vacuum pressure leak detection
- Battery voltage
- Buoyancy pump
- Air bladder inflation
- Iridium modem
- Telemetry
- GPS receiver

NOTE *When starting a mission by using a command from a connected PC, or running only the self test from a PC, a single beep will sound before the test, and either a single beep for pass or ten beeps in succession for fail will sound after completion of the test. It may be difficult or not possible to hear these beeps in a noisy environment, such as on the deck of a ship. If possible, it is best to listen for the beeps in a relatively quiet environment while pressing your ear against the housing below the damper disk.*

2.3.1 Starting the Mission by Deploying the Float Directly from its Crate

The mission will start, beginning with a self test, after the float is deployed directly from its crate and it descends to or below its pressure activation depth. For instructions on how to deploy the float, refer to “Deploying the Float” on page 2-8.

2.3.2 Starting the Mission using the Reset Tool

To start the mission using the Reset tool:

1. Slowly swipe the Reset tool against and across the housing beginning about five centimeters *above* the RESET label and ending in a semicircle about five centimeters below the RESET label. This label is shown Figure 2-4.

   After about 30 seconds, a beep will sound for about three seconds, and then the float will perform the self test which includes inflating the air compartment and filling the oil compartment of the bladder. The self test typically takes about

25 minutes to complete. If all of the checks performed by the self test pass, the beep will sound again for about three seconds, files will be uploaded and downloaded over the Iridium satellite network, and at the end of the Mission Prelude phase, the mission will start. For information on the Mission Prelude phase, refer to SECTION 3: "Operating Modes." Should any one or more of the checks fail, ten short beeps will sound to indicate that the self test failed. The float should not be deployed if the self test fails.

Figure 2-4: RESET Label on Float Housing

2. Check that the air compartment of the bladder is deflated by temporarily removing the large cowling plug and verifying that you can insert your finger inside the hole.

CAUTION *Once the float starts running a mission, do not swipe the reset tool again, as doing so may stop the mission. However, to intentionally stop the mission, connect a PC to the float and start a terminal program as described in "Connecting a PC to the Float" on page 2-10, and then enter sys_reboot.*

2.3.3 Starting the Mission with a Command from a Connected PC

To start the mission with a command from a connected PC:

1. Connect a PC to the float and start a terminal program as described in "Connecting a PC to the Float" on page 2-10.
2. Enter `m_state`. The float should respond as in the following example:

```
05/07/20 20:25:51 2.14.6 STD IRID Final
Onboard Datetime:                  06/01/2020  16:22:32 UTC
Mission State:                     IDLE
Time in State:                     0000:00:01
Activation Pressure:               25.00 dbar
Standby Mode:                      ON
Console Verbosity:                 0
```

   Mission State must be IDLE. If Mission State is not IDLE, enter `sys_reboot`, and then enter `m_state` again and verify that Mission State is IDLE.

3. While listening to the float, enter `m_deploy`.

   A beep will sound for about three seconds, and then the float will perform the self test which includes inflating the air compartment and filling the oil compartment of the bladder. The self test typically takes about 25 minutes to complete. Progress and diagnostic messages will be displayed on the PC as described in "Self Test Progress and Diagnostic Messages" on page 2-8. If all of the checks performed by the self test pass, the beep will sound again for about three seconds, files will be uploaded and downloaded over the Iridium satellite network, and at the end of the Mission Prelude phase, the mission will start. For information on the Mission Prelude phase, refer to SECTION 3: "Operating Modes." Should any one or more of the checks fail, ten short beeps will sound to indicate that the self test failed. The float *should not* be deployed if the self test fails.

4. Verify that the cowling bottom is slightly expanded during the test, indicating bladder inflation.

5. Disconnect the PC from the float.

   To abort the mission, reconnect and enter `sys_reboot`.

2.3.4 Running the Self Test from a Connected PC

When running the self test from a connected PC, the mission will not start. To run the self test from a connected PC:

1. Connect a PC to the float and start a terminal program as described in "Connecting a PC to the Float" on page 2-10. Once the connection is established the float must be placed into Console mode as described in "Console Mode" on page 2-12.

2. Enter `m_state`. The float should respond as in the following example:

```
05/07/20 20:25:51 2.14.6 STD IRID Final
Onboard Datetime:                  06/01/2020  16:20:45 UTC
Mission State:                     CONSOLE
Time in State:                     0000:00:15
Time Remaining in State:           0001:59:45
Activation Pressure:               25.00 dbar
Standby Mode:                      OFF
Console Verbosity:                 5
```

3. While listening to the float, enter `sys_self_test`.

   A beep will sound for about three seconds, and then the float will perform the self test which includes inflating the air compartment and filling the oil compartment of the bladder. The self test typically takes about 25 minutes to complete. Progress and diagnostic messages will also be displayed on the PC as described in "Self Test Progress and Diagnostic Messages" below. If all of the checks performed by the self test pass, the beep will sound again for about three seconds. Should any one or more of the checks fail, ten short beeps will sound to indicate that the self test failed. The float should not be deployed if the self test fails.

Note also that the cowling bottom slightly expands, and then after the test, deflates. This process happens relatively quickly.

4. Enter `m_bye` and then disconnect the PC from the float.

2.3.5 Self Test Progress and Diagnostic Messages

When starting the mission from a PC or running the self test from a PC, progress and diagnostic messages will be displayed on the PC as in the following partial example:

```
> sys_self_test
20200604T200241|5|T_CMD|sys_self_test


********************************************************************************
System Self Test Started @ 2020-06-04 20:02:43
********************************************************************************

Float ID: f1234
Firmware Version: 03/05/20 14:04:09 2.14.3 STD Final
-------------------------------------Vitals-------------------------------------
Battery Voltage:     15.13 V
Float Current:       33.0490 mA
Coulomb Count:       242.9231 mAh
Bladder Pressure:    8.0 dbar
Internal Vacuum:     8.1 dbar
Buoyancy Position:   262 counts
--------------------------------------------------------------------------------
```

For the full example text, refer to APPENDIX H: “Diagnostic Messages Example.”

2.4 Deploying the Float

A float is ready for deployment directly from its crate after preparing it for deployment as described in “Preparing the Float for Deployment” on page 2-3.

NOTE *If a float has been in storage for an extended period of a year or more, a self test from a connected PC should be run to obtain the battery voltage and coulomb counter values. This information should be provided to Teledyne Webb Research customer service before deploying the float to verify that the battery capacity is adequate for the intended mission.*

NOTE *If the mission is started by using the Reset tool or a command from a connected PC, the float should be deployed after the self test is complete but before the end of the Mission Prelude phase which by default is 120 minutes, otherwise unexpected results may occur.*

Typically, floats are deployed from a stationary or moving vessel. They can also be air deployed. For instructions on how to air deploy a float, contact Teledyne Webb Research customer service.

Figure 2-5: Hole in Damper Ring

To deploy the float:

1. Pass a line through the hole in the damper ring. This hole is shown in Figure 2-5. The line should be able to easily slip through the hole and support the weight of the float which is approximately 25 kg (55 lb).

2. While holding both ends of the line, carefully lower the float into the water as shown in Figure 2-6. Do not let the line continuously slide through the hole while lowering the float, as it may cut through it.

CAUTION *When deploying the float, be careful not to let the CTD or other sensors or the antenna strike the side of the ship, as doing so could damage these components.*

Figure 2-6: APEX Profiling Float being Deployed

3. When the float is in the water, let go of one end of the line and carefully pull the other end until the entire line passes through the hole and the float is free. Do not leave the line attached to the float.

   If the float was deployed directly from its crate or after performing a self test from a connected PC, it will float on its side until the cowling fills with water. Then it will submerge and begin descending. During its descent it will periodically check to see if it is at or below its pressure activation depth. When it reaches or passes this depth, it will begin the self test. The actual depth at which the test begins could be well below the pressure activation depth depending on how often the float checks its depth. During the self test, the float will surface, obtain its GPS position, perform time synchronization of the internal clock with the GPS satellite time if needed, and upload log files and download configuration files to and from the server, respectively, over the Iridium satellite network. The mission will then start with the descent to Park depth.

   If the float was deployed after using the Reset tool or a command from a connected PC to start the mission, the mission will start with the Mission Prelude phase. During this phase, the float will run the internal self test to obtain its GPS position, perform time synchronization of the internal clock with the GPS satellite time if needed, and upload and download files over the Iridium satellite network. When the Mission Prelude phase is complete, the float will descend to the Park depth.

2.5 Recovering the Float

Before recovering the float, it must be placed into recovery mode as described in “Recovery Mode” on page 3-23. When recovering the float be careful not to damage the CTD or the antenna.

2.6 Connecting a PC to the Float

To communicate with the float from a PC, you must connect a PC to the float using the RS-232 Current Loop Converter, and the PC must be running a terminal program, such as Procomm Plus, Tera Term or HyperTerminal. It is recommended to set the terminal program to automatically log the console output with timestamping enabled. The PC must also have an RS-232 serial port that uses either a 25-pin or a 9-pin connector. A serial port adapter is included with the RS-232 Current Loop Converter for a PC that has only a 9-pin serial port connector. In addition, the settings for the serial port must be as follows:

Baud Rate:	19200
Parity:	None
Data Bits:	8
Stop Bits:	1
Flow Control:	None
Handshaking:	None
Duplex:	Full

Figure 2-7: Communications Cable Connected to Zinc Anode and Terminal Post (Pressure Sensor)

To connect a PC to the float:

1. Connect the clamps of the communications cable from the RS-232 Current Loop Converter to the zinc anode and to the terminal post (pressure sensor on some floats) on the upper end cap of the float as shown in Figure 2-7. It does not matter which clamp connects to which part.
2. Connect the 25-pin connector of the RS-232 Current Loop Converter to the RS-232 serial port of the PC. Use the serial port adapter if the serial port of the PC uses a 9-pin connector.
3. Attach the appropriate AC adapter plug for the AC outlet to be used to the 15 VDC Power Supply. To attach the plug, push the button labeled “PUSH” on the power supply, slide out the transparent cover, and then slide in the adapter plug.
4. Plug the 15 VDC Power Supply into the AC outlet.
5. Turn on the PC and start the terminal program.
6. Press the Enter key.

   After several seconds the command prompt ">" will be displayed.

NOTE *With the PC connected to the float, all entries made by the float to the system_log.txt file will be displayed along with the operator entries and float responses in accordance with the verbosity level. The verbosity level is set with the System user command **sys_verbosity**. For information on all the available user commands, refer to “User Commands” on page 5-5. For information on the system_log.txt file, refer to “Log Files” on page 4-17.*

If the float does not respond, try one or more of the following:

- Check the serial port settings as listed in the previous page.
- Check the connections to the float.
- Be sure the float is not wet.
- Reboot the PC and restart the terminal program.
- Contact Teledyne Webb Research customer service.

2.7 Console Mode

After the PC is connected to the float for operator intervention versus starting a mission, the float must be placed into Console mode by entering `m_console` at the user prompt. Console mode disables lowpower standby operation and sets the float verbosity level to 5 for displaying output from the float. Entering `m_bye` exits Console mode and re-enables lowpower standby operation. After two hours of no operator input, the float will *automatically* exit Console mode, re-enable lowpower standby operation, set the verbosity to 0, and revert to the Mission State of IDLE.

2.8 Enabling and Disabling Lowpower Standby Operation

Lowpower standby operation is enabled ("on") when a float is shipped. When enabled, the float will remain in a lowpower state during periods when it is not active, drawing very little power from the batteries. When communicating with the float from a PC as described in "Connecting a PC to the Float" on page 2-10, lowpower standby operation can be disabled ("off") by entering `m_console`. Before *disconnecting* the PC, lowpower standby operation should be re-enabled by entering `m_bye`. However, the float will automatically re-enable lowpower standby operation two hours after last entering `m_console`, whether connected to a PC or not.

2.9 Pre-deployment Testing

All floats are shipped from the factory fully tested and ready to deploy. However, damage could occur during transit to the storage or deployment location. To maintain the warranty on the float, Teledyne Webb Research requires that a self test be performed on the float prior to deployment to ensure that all hardware devices are functioning properly as described in to APPENDIX H: "Diagnostic Messages Example." The test results are recorded in the System Log file and must be provided to and verified by Teledyne Webb Research prior to deployment.

For instructions on how to connect a PC to the float and start a terminal program, refer to "Connecting a PC to the Float" on page 2-10. At the command prompt, enter `m_console` to place the float into Console mode and automatically set the verbosity to 5 to enable output of progress messages. The float will time out and exit Console mode in two hours and set the verbosity back to 0. Or before disconnecting the PC from the float, enter `m_bye` to exit Console mode if you are finished with the pre-deployment tests before that time. Entering `m_bye` will set the verbosity back to 0.

2.9.1 Testing the Float

Diagnostic tests can be run on the float, including the self test and the communications test. In addition, a simulated mission can be run out of the water. The communications test is part of the self test but can be run independently.

Running the self test. To run the self test bring the float outside and place it where it has a clear view of the sky for Iridium satellite communications and enter `sys_self_test`, or leave the float indoors and enter `sys_self_test -nosky`. Progress and diagnostic messages should be displayed on the PC as described in "Self Test Progress and Diagnostic Messages" on page 2-8.

Running the communications test. To run the communications test, enter `modem_test`. The Iridium modem's primary and secondary numbers are tested and the results are displayed. Each number can be independently tested as described in "Modem Commands" on page 5-8.

Running a simulated mission. To run a simulated mission out of the water for a float with a pumped type CTD that *does not* have a pH sensor, first install a plastic water loop filled with deionized water and then enter `m_deploy`. For a float with a non-pumped type CTD, just enter `m_deploy`. The float will cycle through the mission phases based on the various timeout settings of the configuration parameters instead of the water depths while displaying progress messages.

CAUTION *Do not use deionized water when running a simulated mission for a float with a CTD-plus-pH sensor. Instead, use UV filtered natural seawater. Use of deionized water with a CTD-plus-pH sensor will affect the calibration of the pH sensor resulting in inaccurate data.*

NOTE *Continuous sampling will not run for floats with a pumped type CTD at the surface, as the "pcutoff" pressure of the CTD is typically set to 2 decibars.*

2.9.2 Viewing and Editing the Configuration Parameters

The mission, sample, system, and sensor configuration parameters can be viewed and edited directly from the PC and edited remotely over the Iridium satellite network.

Viewing the configuration parameters. To view the configuration parameters, enter `fs_cat` followed by a space and the configuration parameter file name: mission.cfg, sample.cfg, system.cfg, or sensor.cfg.

Editing the mission configuration parameters. To edit the mission configuration parameters directly from the PC, refer to “Modifying the mission.cfg File Configuration Parameter Settings Locally from a Connected PC” on page 5-1 and begin with Step 2. To edit the mission configuration parameters remotely over the Iridium satellite network, refer to “Modifying the mission.cfg File Configuration Parameter Settings Remotely” on page 5-3 and then refer to “Transferring Files” on page 2-15 to download the file from the server to the float over the Iridium satellite network.

Editing the sample configuration parameters. To edit the sample configuration parameters remotely over the Iridium satellite network, refer to “Modifying the sample.cfg File Configuration Parameter Settings Remotely” on page 5-4 and then refer to “Transferring Files” below to download the file from the server to the float over the Iridium satellite network.

To edit the sample configuration parameters directly from the PC:

1. Enter `fs_mv sample.cfg sample.old` to rename the current sample.cfg file to sample.old.
2. Enter `sys_capture sample.new` to create a new file called sample.new.
3. Enter the sampling behavior in accordance with “sample.cfg File” on page 4-2.
4. Type "." and then press Enter to exit capture.
5. Enter `fs_cat sample.new` to display the contents of the sample.new file.
6. Enter `fs_mv sample.new sample.cfg` to rename the new file to sample.cfg for use by the float.
7. Enter `sys_reboot` to reset the float and load the new sample configuration.
8. Enter `m_bye` to reset the float with the new sample.cfg file. This command will also exit Console mode.

NOTE *To continue with other pre-deployment tests, you must re-enter m_console to place the float back into Console mode.*

Editing the system configuration parameters. The system configuration parameters should not be modified. However, for the RUDICS/PSTN configuration, to edit the username, password or dial strings, enter `iridium_dial_config` *<username>* *<password>* *<primary_dial_string>* `dialup |rudics` *<alternate_dial_string>* `dialup | rudics`, where "|" indicates a choice of one of the two parameters.

After modifying the RUDICS/PSTN configuration, run the **modem_test** Modem command as described in “Modem Commands” on page 5-8.

Editing the sensor configuration parameters. The sensor configuration parameters should not be modified.

2.9.3 Transferring Files

The mission.cfg and sample.cfg files can be downloaded over the Iridium satellite network from the server to the float. Similarly, the log files can be uploaded from the float to the server. When transferring the files, place the float where it has a clear view of the sky.

Downloading mission.cfg and sample.cfg files. To download the mission.cfg and sample.cfg files from the server to the float, enter `modem_transfer`.

Uploading the log files. To upload the log files from the float to the server, enter `log_up`, and then enter `modem_transfer`.

SECTION 3: Operating Modes

The mission of a Teledyne Webb Research APEX profiling float can start in one of three ways: the float is deployed directly from its crate and descends to or below its pressure activation depth at which time it activates, the Reset tool is swiped across the RESET label on the housing, or a command is issued from a connected PC. In all cases the float has four main operating modes:

- Idle mode
- Mission mode
- Recovery mode
- Emergency mode

The float is shipped in Idle mode, and Recovery mode is activated only by the operator, and typically only if the float is to be recovered. The behavior of the float during each mode is determined by configuration parameters which are in a file called mission.cfg. This file is preloaded onto a secure digital (SD) card in the float but can be modified by the operator as described in SECTION 5: "Modifying the Mission Plan." Figure 3-1 shows a graphical representation of the operating modes, their associated mission.cfg configuration parameters and their factory default settings which are based on a float that is ballasted for a 2000-meter Deep Descent phase following a Park phase of 1000 meters for a 10-day mission duration.

NOTE *Along with the configuration parameters described in this section, there are additional configuration parameters in the mission.cfg file that are specific to the Ice Avoidance function as described in APPENDIX B: "Ice Avoidance." In addition, the* ***TelemetryInterval****,* ***PreludeTime****,* ***UpTime*** *and other configuration parameters, as described in this section, have different settings or function differently for the optional Argos telemetry link as described in APPENDIX D: "Argos Telemetry."*

3.1 Idle Mode

In Idle mode the float will periodically awaken from the lowpower state and ensure the air compartment of the bladder is deflated and the buoyancy position is in accordance with **MActivationCount** described below. In Idle mode the float will also check for the pressure activation depth indicating that it is underwater. When at or below this depth, the float will activate and go into Mission mode.

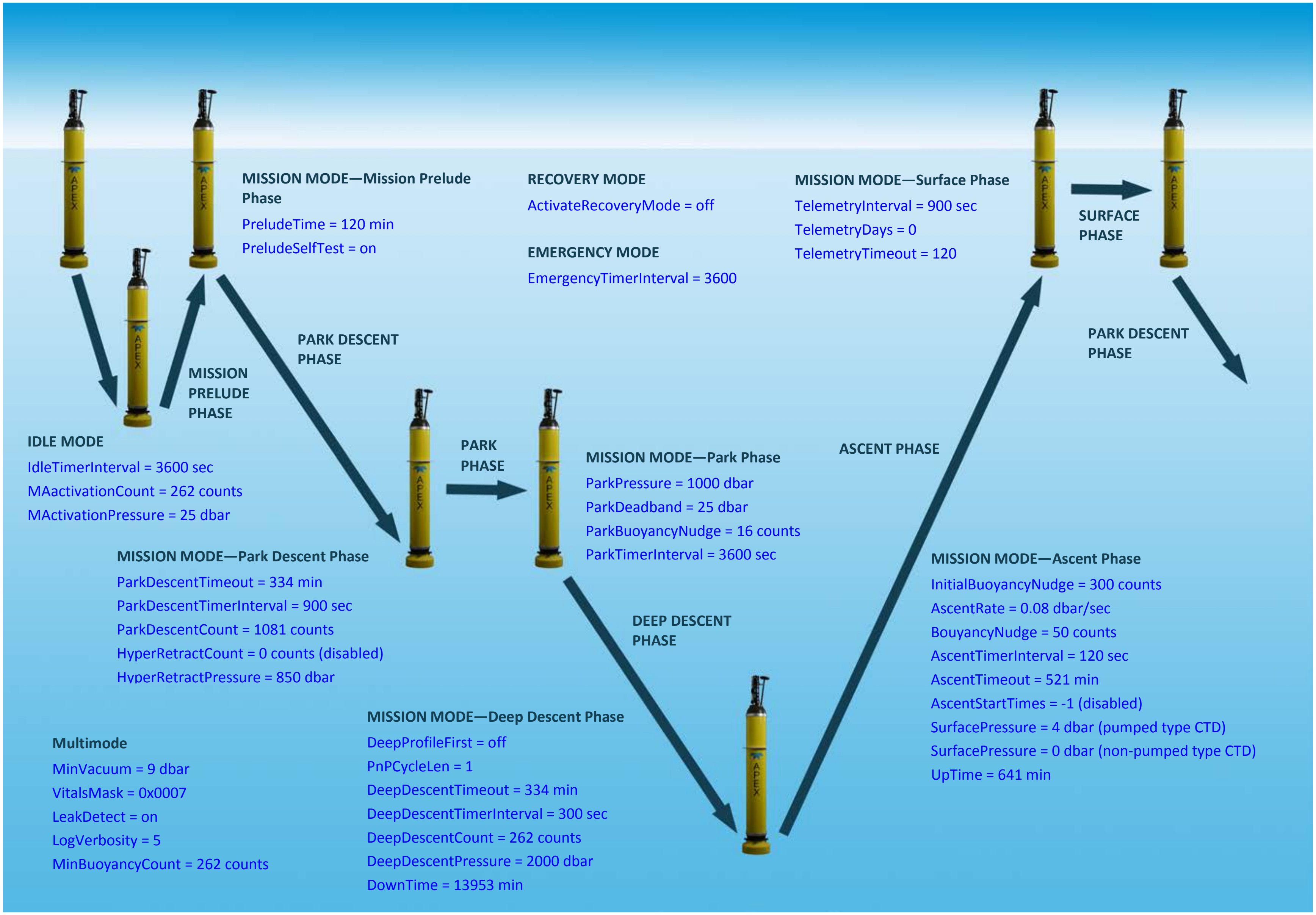


Figure 3-1: Operating Modes and Factory Default mission.cfg File Configuration Parameter Settings

The mission.cfg file configuration parameters that apply to the Idle mode are **IdleTimerInterval**, **MActivationCount** and **MActivationPressure**:

IdleTimerInterval. The time in seconds between which the float momentarily awakens from the lowpower state to verify the buoyancy position and check the pressure. If the pressure is more than **MActivationPressure**, the float goes into Mission mode.

Syntax:	IdleTimerInterval <*seconds*>
Range:	5–86400 sec (24 hours)
Default:	3600 sec (1 hour)

MActivationCount. The buoyancy position in counts that will cause the float to descend below its specified pressure activation depth as determined by **MActivationPressure** when it is deployed.

Syntax:	MActivationCount <*counts*>
Range:	**buoyancy_pump_min – buoyancy_pump_max** counts
Default:	262 counts

For information on **buoyancy_pump_min** and **buoyancy_pump_max**, refer to “system.cfg File” on page 4-14.

CAUTION *The float will not pressure activate if* ***MinBuoyancyCount*** *is greater than* ***MActivationCount****, as* ***MinBuoyancyCount*** *would prevent the buoyancy position from reaching* ***MActivationCount*** *which is associated with* ***MActivationPressure****. For information on* ***MinBuoyancyCount****, refer to “Multimode Mission Configuration Parameters” on page 3-24. It will also not pressure activate if* ***MActivationCount*** *does not allow the float to reach* ***MActivationPressure****.*

NOTE *Although an* ***MActivationCount*** *that is less than* ***DeepDescentCount*** *is valid, when descending for pressure activation, the float may go deeper than the Profile depth as determined by* ***DeepDescentPressure****. For information on* ***DeepDescentCount*** *and* ***DeepDescentPressure****, refer to “Deep Descent Phase” on page 3-9.*

MActivationPressure. The minimum pressure in decibars at which the float goes into Mission mode. It determines the float’s pressure activation depth. The float checks for this pressure periodically at the rate specified by **IdleTimerInterval**.

Syntax:	MActivationPressure <*decibars*>
Range:	5–25 dbar
Default:	25 dbar

3.2 Mission Mode

In Mission mode the float is directed through six distinct phases, five of which are repeated in a continuous cycle:

MISSION PRELUDE: When the float reaches or passes its pressure activation depth, it performs the self test to ensure it is ready to begin the deployment, and depending on the allowed time for this phase, the float may perform a self test to ensure it is ready for deployment, or it may begin the next phase, the Park Descent phase. The Mission Prelude phase occurs first and only once in a deployment.

PARK DESCENT: The float descends to its Park depth of typically 1000 meters and may collect samples or record profiling data when specified. Before starting the descent, an atmospheric pressure measurement is made to check the pressure sensor calibration.

PARK: The float drifts for a configured period while dynamically maintaining its Park depth and may collect samples when specified.

DEEP DESCENT: The float descends to its Profile depth and may collect samples or record profiling data when specified. Upon reaching this depth, which is typically 2000 meters, the float immediately begins its Ascent phase.

ASCENT: The float ascends to the surface at a configured ascent rate while collecting samples and recording profile data. At a configured depth just below the surface, the float begins its Surface phase.

SURFACE: The float transmits its data and position, performs time synchronization of the internal clock on the Controller board with the GPS satellite time if needed, and receives new mission.cfg and sample.cfg files if available. The float may also collect samples or perform measurements while at the surface when specified. Following this phase is a new profiling cycle which begins with the Park Descent phase.

3.2.1 Mission Prelude Phase

The Mission Prelude phase starts when Mission mode starts which happens with pressure activation, a swipe of the Reset tool or a command from a connected PC. During the Mission Prelude phase, the float performs the self test and surfaces if it is not already at the surface, and depending on the time allowed for this phase, transmits all of its log files and receives new mission and sample configuration parameters, if available, over the Iridium satellite network. In addition, it acquires and transmits its position and performs time synchronization and GPS Almanac updates if needed. At the end of the Mission Prelude phase, the float transitions to the Park Descent phase if the self test passed. For a list of tests performed during the self test, refer to “Starting the Mission” on page 2-4.

The mission.cfg file configuration parameters that apply to the Mission Prelude phase are **PreludeTime** and **PreludeSelfTest**.

PreludeTime. The amount of time in minutes the float will remain in the Mission Prelude phase before transitioning to the Park Descent phase. The float will transition to the Park Descent phase *only* at the end of this period and *only* if the self test passes or **PreludeSelfTest** is set to "off." A setting of 0 causes the self test to be skipped. The recommended setting is 60 or higher to ensure the self test is completed and the communications tasks are performed.

Syntax:	PreludeTime <*minutes*>
Range:	0, 1–600 min (10 hours)
Default:	120 min (2 hours)

PreludeSelfTest. If set to "on," the float will transition to the Park Descent phase at the end of the period determined by **PreludeTime** *only* if the self test passes. If the self test fails, the float will go into Emergency mode. If set to "off," the float will still run the self test and then transition to the Park Descent phase whether or not the self test passes. In addition, the test results, pass or fail, will be logged to the system_log.txt file. For information on the system_log.txt file, refer to “Log Files” on page 4-17.

Syntax:	PreludeSelfTest <on|off>
Range:	on or off
Default:	on

3.2.2 Park Descent Phase

The Park Descent phase begins with a descent from the surface by both deflating the air compartment of the bladder and applying the initial Park Descent buoyancy adjustment which extracts a configured amount of oil from the oil compartment of the bladder. With the buoyancy adjusted, the float begins its descent to its Park depth and transitions to the Park phase upon reaching the Park depth or if it times out attempting to reach it. Periodic or continuous sampling may be peformed as the float descends as described in “Mission Phase Sampling” on page 3-21. Should the float detect that it has reached the bottom during the Park Descent phase, it will transition to the Park phase.

The mission.cfg file configuration parameters that apply to the Park Descent phase are **ParkDescentTimeout**, **ParkDescentTimerInterval**, **ParkDescentCount, HyperRetractCount**, and **HyperRetractPressure**:

ParkDescentTimeout. The amount of time in minutes allowed for the float to leave the surface and to reach its Park depth. Should the float not reach its Park depth in this period, it will transition to the Park phase. The default setting is 334 minutes which is based on a descent from the surface to a depth corresponding to a pressure of 1000 dbar at a rate of 0.05 dbar/sec.

Syntax:	ParkDescentTimeout <*minutes*>
Range:	1–1380 min (23 hours)
Default:	334 min

ParkDescentTimerInterval. The interval in seconds between which the float awakens from the lowpower state to sample the pressure to determine if the Park depth has been reached or to sample the CTD and the optional sensors. It is also the interval at which the sensor specified in a sample.cfg entry for the Park Descent phase that includes an interval parameter setting of 0 is sampled. Refer to “sample.cfg File” on page 4-2 for more information on the sample.cfg file.

This setting should be made as long as possible to conserve energy. It is also the maximum amount of time the float will remain uninterrupted in the lowpower state during the Park Descent phase. The float will wake up earlier if it determines that the Park depth will be reached before the end of the next interval or to perform sampling. The pressure is recorded to the science_log.bin file. For information on the science_log.bin file, refer to “Log Files” on page 4-17.

Syntax:	ParkDescentTimerInterval <*seconds*>
Range:	5–5400 sec (90 minutes)
Default:	900 sec (15 minutes)

ParkDescentCount. The buoyancy position in counts that will cause the float to descend to its Park depth. This configuration parameter is dynamically updated during the Park Descent and Park phases of each profiling cycle in accordance with **ParkBuoyancyNudge** of the Park phase and is applied during the next profiling cycle.

Syntax:	ParkDescentCount *<counts>*
Range:	**DeepDescentCount** – **buoyancy_pump_max** counts
Default:	1081 counts

NOTE *A float will not reach the Park depth if **ParkDescentCount** is less than **MinBuoyancyCount**. In addition, **ParkDescentCount** must not be set less than **DeepDescentCount**.*

NOTE *The default **ParkDescentCount** is based on the default **ParkPressure** of 1000 dbar. For information on **ParkPressure**, refer to "Park Phase" below.*

HyperRetractCount. The buoyancy position in counts at the start of the Park Descent phase. This setting is typically less than **ParkDescentCount**. A setting of 0 disables the hyper retract operation. When enabled, if **HyperRetractPressure** will not be reached within **ParkDescentTimeout**, **HyperRetractCount** is adjusted accordingly to extract more oil from the oil compartment of the bladder for the current and subsequent profiling cycles.

Syntax:	HyperRetractCount *<counts>*
Range:	**MinBuoyancyCount** – **buoyancy_pump_max** counts
Default:	0 counts (disabled)

HyperRetractPressure. The pressure at which the buoyancy position is set to **ParkDescentCount** when **HyperRetractCount** is enabled.

Syntax:	HyperRetractPressure *<decibars>*
Range:	0–2000 dbar
Default:	850 dbar

NOTE ***HyperRetractPressure** should be set lower than **ParkPressure**.*

For information on the hyper retract operation, refer to APPENDIX E: "Hyper Retract/N2 Compensation."

3.2.3 Park Phase

The Park phase begins immediately after the completion of the Park Descent phase. Periodic or continuous sampling may be peformed during the Park phase as described in "Mission Phase Sampling" on page 3-21. The Park phase continues for a period determined by the actual descent time to the Park depth and **DownTime** and **DeepDescentTimeout** of the Deep Descent phase as follows:

DownTime – *Actual Park Descent Time* – **DeepDescentTimeout**

Should the float not reach the Park depth before the period specified by **ParkDescentTimeout**, it will transition to the Park phase, and the Park phase period will instead be:

DownTime – ParkDescentTimeout – DeepDescentTimeout

There are two other configuration parameters, **DeepProfileFirst** and **PnPCycleLen**, that will impact the time in the Park phase. These configuration parameters are described in "Deep Descent Phase" on page 3-9. The float can be configured to quickly transition through the Park phase and start a descent to the Profile depth if a descent to the Profile depth is required for the first profiling cycle, or the float can be configured to omit consecutive descents to Profile depth thus staying in Park longer before doing an ascent. If the float reaches the bottom during the Park Descent phase, the float will not dynamically adjust its buoyancy to be within a deadband that is below the ocean bottom.

The mission.cfg file configuration parameters that apply to the Park phase are **ParkPressure**, **ParkDeadBand**, **ParkBuoyancyNudge** and **ParkTimerInterval**:

ParkPressure. The pressure in decibars associated with the Park depth. During the initial Park Descent phase, the float must reach this pressure to be at the Park depth regardless of the setting of **ParkDeadBand**.

Syntax:	ParkPressure *<decibars>*
Range:	0–2000 dbar
Default:	1000 dbar

ParkDeadBand. The maximum pressure difference in decibars between the actual pressure and **ParkPressure**. When exceeded, the buoyancy position is adjusted by **ParkBuoyancyNudge** to move the float towards the Park depth. **ParkDeadBand** should not be set too low, as doing so may result in continual buoyancy adjustments and higher energy usage.

Syntax:	ParkDeadBand *<decibars>*
Range:	1–2000
Default:	25 dbar

ParkBuoyancyNudge. The number of counts to change the buoyancy position from its current position in the direction that will bring the float closer to the Park depth. It is applied after three successive pressure samples taken based on **ParkTimerInterval** that are outside the dead band specified by **ParkDeadBand**.

Syntax:	ParkBuoyancyNudge *<counts>*
Range:	0–2000 counts
Default:	16 counts

ParkTimerInterval. The interval in seconds between float samples of the pressure to determine if the pressure is within the dead band specified by **ParkDeadBand**. It is also the interval at which the sensor specified in a sample.cfg file entry for the Park phase is sampled, but only when the sampling interval is 0, otherwise sampling will occur at the intervals specified in the sample.cfg file. Refer to “sample.cfg File” on page 4-2 for more information on the sample.cfg file. The pressure is recorded to the science_log.bin file. For information on the science_log.bin file, refer to “Log Files” on page 4-17.

Syntax:	ParkTimerInterval *<seconds>*
Range:	5–86400 sec (24 hours)
Default:	3600 sec (1 hour)

3.2.4 Deep Descent Phase

The Deep Descent phase begins with the nearly total extraction of the oil from the oil compartment of the bladder and the descent to the Profile depth. Similar to the Park Descent phase the float will check the descent rate to ensure it is traveling fast enough to reach **DeepDescentPressure** within **DeepDescentTimeout** described below. However, unlike during the Park Descent phase, the float does not remain at **DeepDescentPressure**. It will also over time reduce the amount of oil that is required to get to the Profile depth to conserve energy. Periodic or continuous sampling may be peformed during the Deep Descent phase as described in “Mission Phase Sampling” on page 3-21. Should the float detect that it has reached the bottom during the Deep Descent phase. It will transition to the Ascent phase.

NOTE *The float will end the Deep Descent phase and start the Ascent phase if after three buoyancy adjustments the float has stopped descending and therefore has reached the bottom. Each buoyancy adjustment occurs in accordance with **DeepDescentTimerInterval** described below.*

The mission.cfg file configuration parameters that apply to the Deep Descent phase are **DeepProfileFirst**, **PnPCycleLen**, **DeepDescentTimeout**, **DeepDescentTimerInterval**, **DeepDescentCount**, **DeepDescentPressure**, and **DownTime**:

DeepProfileFirst. If set to "on," the float will, on its first descent only, after the Mission Prelude, descend immediately to the Profile depth after reaching the Park depth, essentially bypassing the Park phase. If set to "off," the first and all subsequent Park phase periods are as described in “Park Phase” on page 3-8.

Syntax:	DeepProfileFirst <on|off>
Range:	on or off
Default:	off

PnPCycleLen. The number of descents the float will make to the Park depth before a descent to the Profile depth. If **DeepProfileFirst** is "on," then **DeepProfileFirst** will run before **PnPCycleLen** applies. A setting of 0 disables the Deep Descent phases.

Syntax:	PnPCycleLen <*n*>
Range:	0, 1–255
Default:	1

DeepDescentTimeout. The amount of time in minutes allowed for the float to leave its Park depth and to reach its Profile depth. Should the float not reach its Profile depth in this period, it will stop descending and transition to the Ascent phase. The default setting is 334 minutes which is based on a descent from the Park depth to a depth corresponding to an increased pressure of 1000 dbar at a rate of 0.05 dbar/sec. This configuration parameter also affects the rate of descent, increasing the rate if necessary to attempt to reach the Profile depth within the allowed time.

Syntax:	DeepDescentTimeout <*minutes*>
Range:	1–1380 min (23 hours)
Default:	334 min

DeepDescentTimerInterval. The interval in seconds between which the float awakens from the lowpower state to sample the pressure to determine if the Profile depth has been reached or to sample the CTD and the optional sensors. It is also the interval at which the sensor specified in a sample.cfg entry for the Deep Descent phase that includes an interval parameter setting of 0 is sampled. Refer to “sample.cfg File” on page 4-2 for more information on the sample.cfg file.

This setting should be made as long as possible to conserve energy. It is also the maximum amount of time the float will remain uninterrupted in the lowpower state during the Deep Descent phase. The float will wake up earlier if it determines that the Profile depth will be reached before the end of the next interval or to perform sampling. The pressure is recorded to the science_log.bin file. For information on the science_log.bin file, refer to “Log Files” on page 4-17.

Syntax:	DeepDescentTimerInterval *<seconds>*
Range:	5–5400 sec (90 minutes)
Default:	300 sec

DeepDescentCount. The buoyancy position in counts that will cause the float to descend to its Profile depth. This configuration parameter is dynamically updated during each profiling cycle based on the adjustments needed to reach the desired Profile depth.

Syntax:	DeepDescentCount *<counts>*
Range:	**MinBuoyancyCount** – **buoyancy_pump_max** counts
Default:	262 counts

NOTE *A float will not reach the Profile depth if* ***MinBuoyancyCount*** *is greater than* ***DeepDescentCount****.*

NOTE *The default* ***DeepDescentCount*** *is based on the default* ***DeepDescentPressure*** *of 2000 dbar.*

DeepDescentPressure. The pressure in decibars associated with the Profile depth.

Syntax:	DeepDescentPressure *<decibars>*
Range:	0–2000 dbar
Default:	2000 dbar

DownTime. The allocated time in minutes for the float to leave the surface, descend to the Park depth and then descend to the Profile depth, or in cases where the float does not perform a descent to the Profile depth, the time to complete the Park phase.

Syntax:	DownTime *<minutes>*
Range:	2–43200 min (30 days)
Default:	13953 min (9.7 days)

3.2.5 Down Time—Park Descent, Park and Deep Descent Phases

The float attempts to complete the descent from the surface to the Park depth and then to the Profile depth in the Down Time as specified by **DownTime**. However, the actual time in the Park Phase and the setting of **DeepDescentTimeout** vary the actual travel time of the float.

An example is shown in Figure 3-2 where the actual Park Descent Time was 930 minutes, **DownTime** is set to 3540 minutes and **DeepDescentTimeout** is set to 500 minutes. Therefore the actual Park Time was 2110 minutes:

3540 – 930 – 500 = 2110 minutes

The actual Deep Descent Time was 483 minutes versus the **DeepDescentTimeout** setting of 500 minutes. Therefore the actual Down Time was 3523 minutes:

930 + 2110 + 483 = 3523 minutes

which is very close to the **DownTime** setting of 3540 minutes.

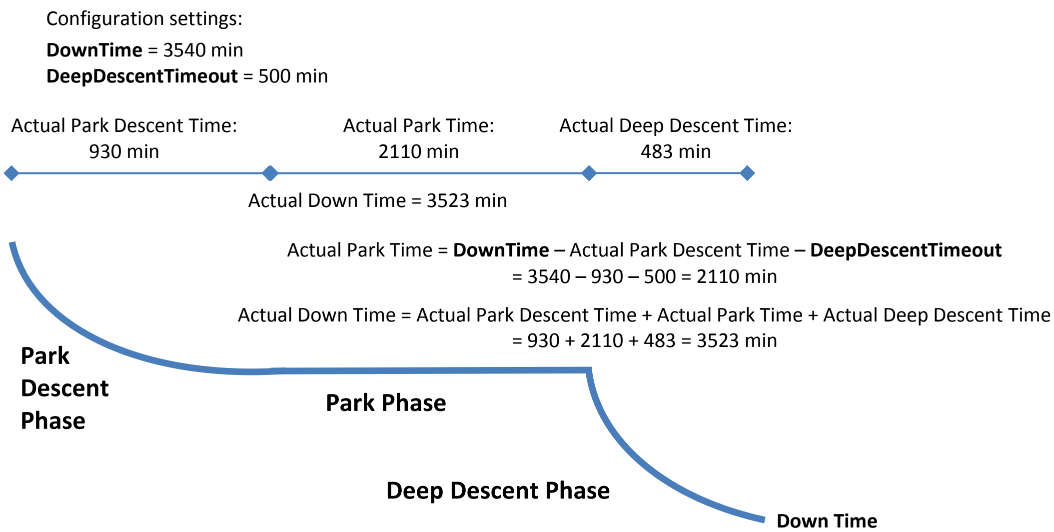


Figure 3-2: APEX Profiling Float Down Time Example

If **DeepDescentTimeout** were to be set *less than* 500 minutes, the float would stay in the Park phase longer and pull in more oil to complete the Deep Descent phase *faster* so as to meet the **DownTime** setting.

If **DeepDescentTimeout** were to be set *more than* 500 minutes, the float would complete the Park phase earlier and pull in less oil to perform the Deep Descent phase *slower* so as to meet the **DownTime** setting.

The effect of these differences in the **DeepDescentTimeout** setting is shown in Figure 3-3.

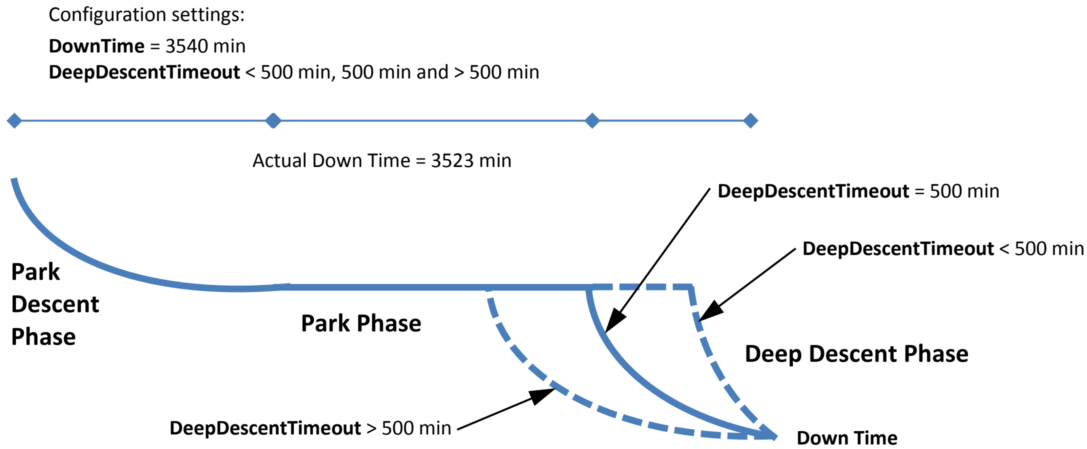


Figure 3-3: APEX Profiling Float Down Time Example—Shorter Versus Longer DeepDescentTimeout Settings

When specifying the Park and Deep Descent depths by setting **ParkPressure** and **DeepDescentPressure**, the following parameters need to be coordinated:

Buoyancy counts. The buoyancy count values are associated with a particular depth. Initially they must be set, but over time the float will dynamically adjust them to reach the specified depth. When a new Park or Deep Descent depth is specified, the buoyancy count for **DeepDescentCount** of the Deep Descent phase and **ParkDescentCount** of the Park Descent phase need to be specified.

Timeouts. The timeout values are associated with a particular depth. When a new depth is specified, the associated timeout values also need to be specified. Both **DeepDescentTimeout** of the Deep Descent phase and **AscentTimeout** of the Ascent phase need to be reviewed. The timeouts also impact the behavior of the float and should be carefully calculated to meet the required behavior.

Mission duration. Changes in depth may impact **DownTime** of the Deep Descent phase and **UpTime** of the Ascent phase. These parameters should be reviewed with respect to new depths.

AscentStartTimes. Changes in depth may impact **AscentStartTimes** of the Ascent phase. This configuration parameter should be reviewed with respect to new depths.

ParkPressure and DeepDescentPressure. Changes made to either or both **ParkPressure** of the Park phase and **DeepDescentPressure** of the Deep Descent phase may impact each other. Both parameters should be reviewed when either is changed.

HyperRetractPressure. When changing **ParkPressure**, **HyperRetractPressure** of the Descent phase may be impacted.

3.2.6 Ascent Phase

Typically the Ascent phase begins immediately after the completion of the Deep Descent phase with the partial filling of the oil compartment of the bladder to provide an initial buoyancy nudge. However, the Ascent phase will begin immediately after the Park phase if **PnPCycleLen** is set to 0. With the bladder partially filled, the float begins its ascent to the surface and starts acquiring profile data. Periodic sampling may be performed at the deeper depths as described in "Mission Phase Sampling" on page 3-21, transitioning to continuous sampling at a configured depth during the ascent until the float is near the surface where the sampling stops. For a float with a non-pumped type CTD, the sampling can continue all the way to the surface.

NOTE *During the Ascent phase the normal behavior of the float will be modified if ice avoidance is enabled and ice is detected. For information on ice avoidance and how to enable it, refer to APPENDIX B: "Ice Avoidance."*

The mission.cfg file configuration parameters that apply to the Ascent phase are **InitialBuoyancyNudge**, **AscentRate**, **BuoyancyNudge**, **AscentTimerInterval**, **AscentTimeout**, **AscentStartTimes**, **SurfacePressure**, and **UpTime**:

InitialBuoyancyNudge. The number of counts to add to the current buoyancy position to start the Ascent phase. **InitialBuoyancyNudge** is also applied when the float reaches the surface prior to establishing satellite communications as described in "Surface Phase" on page 3-16.

Syntax:	InitialBuoyancyNudge *<counts>*
Range:	0–2000 counts
Default:	300 counts

AscentRate. The rate of ascent in decibars per second that the float attempts to maintain. During the ascent the float measures its actual ascent rate every **AscentTimerInterval**, and if it is less than **AscentRate**, the buoyancy position is moved in accordance with **BuoyancyNudge**.

Syntax:	AscentRate *<decibars/second>*
Range:	0.02–0.20 dbar/sec
Default:	0.08 dbar/sec

BuoyancyNudge. The number of counts to add to the current buoyancy position so as to incrementally increase the buoyancy of the float when its ascent rate is below that specified by **AscentRate**. **BuoyancyNudge** also applies when the float reaches the surface and satellite communications has not been established as described in "Surface Phase" on page 3-16.

Syntax:	BuoyancyNudge *<counts>*
Range:	0–2000 counts
Default:	50 counts

AscentTimerInterval. The interval in seconds between float samples of the pressure to check the ascent rate and to determine if the surface has been reached. It is also the interval at which the sensor specified in a sample.cfg entry for the Ascent phase that includes an interval parameter setting of 0 is sampled. Refer to “sample.cfg File” on page 4-2 for more information on the sample.cfg file. The pressure is recorded to the science_log.bin file. For information on the science_log.bin file, refer to “Log Files” on page 4-17.

Syntax:	AscentTimerInterval *<seconds>*
Range:	5–5400 sec (60 minutes)
Default:	120 sec

AscentTimeout. The amount of time in minutes allowed for the float to reach the surface. Should it not reach the surface in this period, it will stop sampling and transition to the Surface phase and fully inflate the air compartment and completely fill the oil compartment of the bladder. The default setting is 521 minutes which is based on an ascent to the surface from a depth corresponding to a pressure of 2000 dbar at a rate of 0.08 dbar/sec plus 25%.

Syntax:	AscentTimeout *<minutes>*
Range:	1–1380 min (23 hours)
Default:	521 min (8 hours, 41 min)

NOTE *When ice avoidance is enabled,* ***AscentTimeout*** *is used in ice cap detection. For information on ice avoidance and how to enable it, refer to APPENDIX B: “Ice Avoidance.”*

AscentStartTimes. Enables time of day operation with the entry of one to as many as four settings. One setting is required; up to three additional settings are optional. The required setting is the minutes after GMT midnight at which an ascent will start after the initial Park phase period. The three optional settings are minutes after GMT midnight at which up to three additional ascents will start on the same day. For these optional settings, the operator must ensure that there is enough time in the day for the float to descend and then ascend at each of the specified times, otherwise the time of an ascent will be extended to the following day. The operator must also take into account **PnPCycleLen** when estimating ascent times. A setting of -1 disables the time of day operation.

Syntax: AscentStartTimes *<minutes1> [<minutes2>] [<minutes3>] [<minutes4>]*

Range: <minutes1> -1, 0–1439
<minutes2–4> 0–1439

Default: -1 (disabled)

For information on the time of day operation, refer to APPENDIX C: "Time of Day Operation."

SurfacePressure. The pressure in decibars added to the most recently measured pressure at the surface. This total is the pressure at which sampling stops for a pumped type CTD. It is also the pressure below which the Surface phase begins.

Syntax: SurfacePressure *<decibars>*

Range: 0–30 dbar

Default: 4 dbar (for a float with pumped type CTD)
0 dbar (for a float with a non-pumped type CTD)

UpTime. The allocated time in minutes for the float to leave the Profile depth, reach the surface and establish satellite communications. If **TelemetryTimeout** of the Surface phase is disabled, then it also includes the time to complete telemetry. The difference in the setting of this configuration parameter and the actual ascent time determines the maximum amount of time at the surface. However, if **TelemetryTimeout** is enabled and satellite communications is established, then the time is fixed to that of the setting of **TelemetryTimeout**. The default setting is 641 minutes which is based on an ascent to the surface from a depth corresponding to a pressure of 2000 dbar at a rate of 0.08 dbar/sec plus 25% and 2 hours at the surface.

Syntax: UpTime *<minutes>*

Range: 1–1440 min (24 hours)

Default: 641 min

3.2.7 Surface Phase

The Surface phase typically begins when the float ascends to the depth determined by **SurfacePressure** of the Ascent phase or when **AscentTimeout** of the Ascent phase has been reached. When the Surface phase begins, an **InitialBuoyancyNudge** is applied and the air compartment of the bladder inflates to ensure the float reaches the surface and the antenna remains well above the surface. An attempt to establish satellite communications is made. If not established, another **BuoyancyNudge** is done every **TelemetryInterval** as described below until the oil compartment of the bladder is completely filled.

When at the surface, and if satellite communications is established, the float begins transmitting data until all of the data have been transmitted or the time specified by **Uptime** is used. If satellite communications is not established, the float will continue to try to establish communications for the remaining **UpTime**. If **TelemetryTimeOut** is enabled, the **UpTime** is used for the amount of time to establish satellite communications and once established, the float will remain at the surface while transmitting data until all of the data have been transmitted or the time specified by **TelemetryTimeOut** described below is used. Periodic or continuous sampling may be peformed during the Surface phase as described in “Mission Phase Sampling” on page 3-21.

When at the surface, the float transmits all of its log files, which includes all the sensor and position data, and receives new mission and sample configuration parameters if available over the Iridium satellite network. It also performs time synchronization of the internal clock on the Controller board with the GPS satellite time if needed and measures the atmospheric pressure to verify the pressure sensor calibration.

If all the data are transmitted, the float will start a new profiling cycle and not wait for the remainder of the time on the surface. Should communications be unsuccessful before the Surface phase ends and there is no more time left on the surface, the float will start a new profiling cycle without transmitting the recorded data. Should this situation occur, the data will be transmitted during the next Surface phase after the most recently acquired data are transmitted until all of the data are transmitted or the Surface phase ends. While at the surface, the float can be placed into Recovery mode as described in “Recovery Mode” on page 3-23". In Recovery mode the float remains on the surface until it is recovered or remotely taken out of Recovery mode.

NOTE *Raw pressure sensor data are recorded to all the log files. When the float is at the surface, a pressure sample is obtained for reference. The pressure difference from zero is recorded to the system_log.txt file as an offset and may be used for post processing of science data at the start of each Park Descent. Although it is always recorded, it will only be updated if satellite communications was established on the previous surfacing. The difference is applied to **SurfacePressure** to ensure that the CTD is properly turned off at the correct depth. The log entry is of the form:*

“5|PARKDESCENT|update_offset|surface pressure offset updated from 0.00 to 0.20 dbar.”

NOTE *During the Surface phase the normal behavior of the float will be modified if ice avoidance is enabled and ice is detected. For information on ice avoidance and how to enable it, refer to APPENDIX B: “Ice Avoidance.”*

The mission.cfg file configuration parameters that apply to the Surface phase are **TelemetryInterval**, **TelemetryDays**, and **TelemetryTimeout:**

TelemetryInterval. The interval in seconds between which the float attempts to communicate over the Iridium satellite network and transmit its position and log files and download new mission and sample configuration parameters. The float continues the attempts until it succeeds or until the end of the period specified by **UpTime** of the Ascent phase or the end of **TelemetryTimeout**, if enabled, after which a new Park Descent phase begins.

Syntax:	TelemetryInterval <*seconds*>
Range:	5–86400 sec (24 hours)
Default:	900 sec (15 minutes)

NOTE *Should communications fail, the reason might be that the float has not actually surfaced. To account for this possibility, the float will push more oil into the oil compartment of the bladder in accordance with **BuoyancyNudge** after each unsuccessful communications attempt to ensure that the float eventually reaches the surface.*

TelemetryDays. The days of the year during which telemetry is attempted during the Surface phase. When the float surfaces on a day that is *not* specified, it does *not* attempt telemetry and immediately begins the Park Descent phase for the next profiling cycle. **TelemetryDays** are specified with a start day and an end day. The start day is the first day of the transmit season for the year. The end day is the last day of the transmit season for the year. A setting of 0 0 disables **TelemetryDays**, and the float attempts telemetry during each Surface Phase.

CAUTION ***TelemetryDays*** *should not be set such that the transmit season is less than 30 days, as doing so may not allow a long enough transmit season to perform telemetry. The transmit season should be set long enough to ensure that the float has an opportunity to perform telemetry.*

Syntax:	TelemetryDays <*start day*> <*end day*>
Range:	0, 1–366 for both <*start day*> and <*end day*>
Default:	0 0 (disabled)
<u>Example 1</u>:	TelemetryDays 100 200 Telemetry is attempted on day 100 through day 200 of the current year during each Surface phase. It will not attempt

telemetry on day 1 through day 99 or day 201 through day 366 (leap year).

Example 2: TelemetryDays 250 100
Telemetry is attempted on day 250 of the current year through day 100 of the following year during each Surface phase.

NOTE *For deployment locations in the Southern Hemisphere, it is expected that **TelemetryDays** would span the summer months occurring at the end of the current year through the beginning of the following year.*

Example 3: TelemetryDays 1 366
Telemetry is attempted on every day of the current year during each Surface phase.

TelemetryTimeout. The amount of time in minutes allowed for the float to complete telemetry after it establishes satellite communications. A setting of 0 disables **TelemetryTimeout**, and the time available is instead determined by the remaining **UpTime** of the Ascent phase.

Syntax: TelemetryTimeOut <*minutes*>
Range: 0, 1–1440 min
Default: 120 min (2 hours)

When multiple log files are stored on the float for transmission, the float will transmit the files in the order of the most recent to the oldest to include sending all science logs first and then alternate between sending system and vitals log files from most recent to oldest. This scenario may happen if the float was unable to transmit on the previous cycle or has been under ice for multiple mission cycles. You can alter the file transmission order and transmit a cross section of the science logs ranging from the newest to the oldest where the float hops through the list of files.

The **modem_file_upload_scheme** command is used to change the default "legacy" transmission priority to the cross section "hop" scheme or vice versa. The syntax, range and default are the following:

Syntax: modem_file_upload_scheme <legacy|hop>
Range: legacy or hop
Default: legacy

When science log transmission hopping is configured, the float will first send the most recent science, system and vitals log files. Next the float will transmit the science log file in the middle of the list. The float will then send every 4^{th}, 8^{th}, 16^{th} science log file and so on until all science log files have been transmitted. After all science logs are transmitted the float will alternate sending the remaining system and vitals logs from newest to oldest. If the transmission time expires at any time during the hopping scheme, the float continues to the next mission cycle. At the next surfacing the float would restart the hopping scheme with the remaining files.

3.2.8 Up Time—Ascent and Surface Phases

The float attempts to complete the ascent to the surface and to complete telemetry within the Up Time as specified by **UpTime**. The **AscentRate** value, the distance traveled from the start of the ascent, the number of telemetry retries, and the time to transmit the data files need to be considered when setting **Uptime**.

An example is shown in Figure 3-4 where the float took 751 minutes to reach the surface and spent 6 minutes at the surface before commencing the next profiling cycle. **UpTime** is set to 1058 minutes, **AscentTimeout** is set to 938 minutes, and **AscentRate** is set to 0.12 dbar/sec. Therefore the actual Up Time is 757 minutes:

751 + 6 = 757 minutes

which is less than the **UpTime** setting of 1058 minutes.

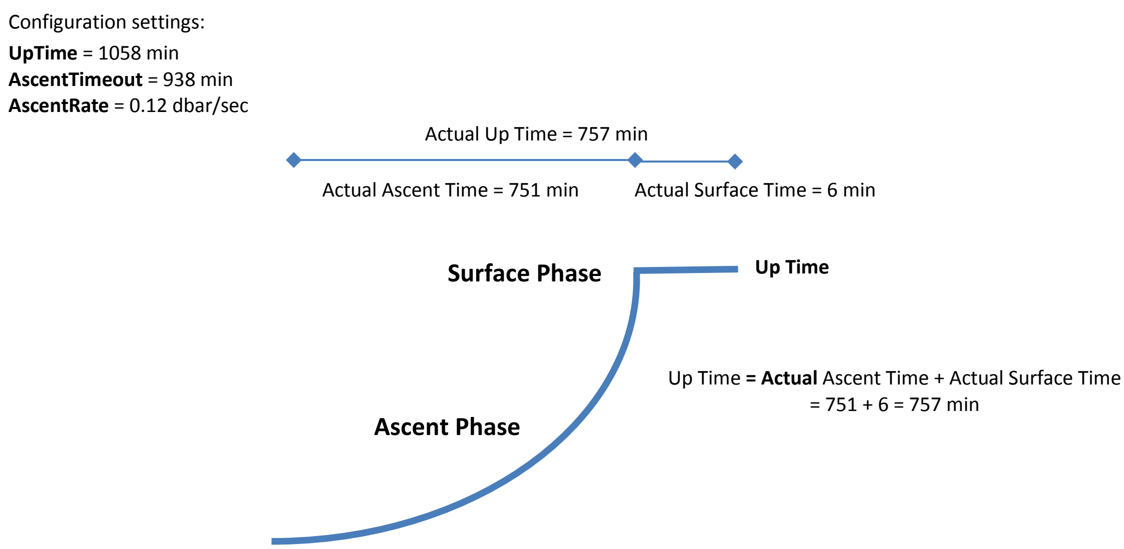


Figure 3-4: APEX Profiling Float Up Time Example

Other factors that influence the actual Up Time of the float include the following:

AscentTimeout. If **AscentTimeout** of the Ascent phase is reached, the float will transition to the Surface Phase. The float will determine the expected time to reach the surface and temporarily increase **UpTime** to ensure that it has enough time at the surface to complete telemetry.

If **TelemetryTimeout** is non-zero, the float will remain at the surface for the specified time while transmitting stored log file until complete or until **TelemetryTimeout** expires. If needed, the time in the Surface phase is temporarily extended after establishing communications. A non-zero **TelemetryTimeout** is recommended to allow the float enough time in the Surface phase to complete telemetry.

TelemetryDays. If **TelemetryDays** is specified, the float will *not* attempt telemetry for those days that are *not* included in **TelemetryDays**. Instead, as soon as the float reaches the surface, it will begin the Park Descent phase.

Number of files to transmit. Typically for each profiling cycle it takes several minutes to transmit the log files. If the float is unable to perform telemetry, it will store the log files and attempt to send them during the next Surface phase. For example, if the float is under ice for several months, once it reaches the surface, it will remain there for the specified **UpTime** or **TelemetryTimeout** while transmitting the remaining files until complete or until **UpTime** or **TelemetryTimeout** ends.

Quality of satellite connections. If the transmission quality is poor, the float will remain at the surface and continue to retry telemetry until either there is a good connection and the files are transmitted or **UpTime** or **TelemetryTimeout** ends.

3.3 Mission Phase Sampling

The float can sample in any one of the mission phases listed below. The Descent and Ascent phases support continuous CTD profiling. Other mission phases, such as Park and Surface, only support interval sampling. The Surface phase sampling occurs around specific events to include air bladder inflation, CTD bin averaging and telemetry. Interval sampling can be done independently or during profiling to collect samples at specified depth or time intervals. Interval samples can include a burst mode for taking a number of consecutive samples at the specified interval. Information on how to configure the float's sampling behavior is described in "sample.cfg File" on page 4-2. The float will also collect samples while adjustments are being made to the buoyancy position.

When sampling with multiple optional sensors, the float will collect samples in the order specified in the sample configuration file. A pressure reading is provided before and after each set of samples collected.

3.3.1 Sampling During the Park Descent Phase

The float can do both continuous CTD profiling during the Park Descent phase and perform interval sampling based on **ParkDescentTimerInterval**, a specified depth interval in decibars or a specified time interval in seconds.

3.3.2 Sampling During the Park Phase

While the float drifts during the Park phase, it can sample based on a specified time interval in seconds or on **ParkTimerInterval** as described in “sample.cfg File” on page 4-24 for interval sampling. The required sampling interval should be specified unless it is the same as **ParkTimerInterval**, where in this case it should be set to 0. When set to 0, the sampling interval *is* **ParkTimerInterval** as described for **ParkTimerInterval** on page 3-9. Furthermore, during the Park phase the float will collect samples only within the depth range specified in the sample.cfg File.

For example, if **ParkTimerInterval** is 3600 seconds, or once per hour, and the required sampling interval is 8 hours, the sampling interval should be set to 28800 seconds. However, if for example, *both* the required sampling interval and **ParkTimerInterval** are 3600 seconds, the sampling interval should be configured to be the same as **ParkTimerInterval** by setting it to 0.

3.3.3 Sampling During the Deep Descent Phase

The float can do both continuous CTD profiling during the Deep Descent phase and interval sampling based on **DeepDescentTimerInterval**, a specified depth interval in decibars or a specified time interval in seconds.

3.3.4 Sampling During the Ascent Phase

The float can do both continuous CTD profiling during the Ascent Descent phase and perform interval sampling based on **AscentTimerInterval**, a specified depth interval in decibars or a specified time interval in seconds. For ice avoidance, the interval temperature samples are used for thermal detection, refer to APPENDIX B: “Ice Avoidance.”

3.3.5 Sampling During the Surface Phase

While the float is in the Surface phase, it can sample based on a specified depth in decibars or a specified time interval in seconds. As the float nears the surface it will turn off the pumped type CTD typically at 4 decibars. The non-pumped type CTD can continue to run until the float breaks the surface and the conductivity becomes 0. The float can continue to collect samples all the way to the surface and will delay CTD bin averaging until it reaches the surface.

Under normal float behavior and conditions, there are four periods where the float will check for an opportunity to collect samples:

- After the float enters the Surface phase prior to running subsurface measurements for the sensors.
- After the surface measurements are complete prior to establishing satellite communications.
- After establishing satellite communications prior to closing the log files for telemetry.
- After telemetry is complete prior to commencing the Park Descent phase.

For each of these periods, if the sampling rules apply for depth or time for the sensor, the float will collect samples. For depth based sampling, the float needs to ascend above the specified depth. For example, if the float is sampling every decibar, it would need to ascend above 3, 2 and 1 decibar between the periods described above to collect samples. For time based interval sampling, the intervals need to have elapsed during the same periods for the sensor to collect samples. Other factors such as ice evasion or telemetry failures would impact the sampling behavior of the float in this near surface area.

3.4 Recovery Mode

Recovery mode is operator activated. If activated, the float remains on the surface indefinitely while transmitting its position and log files and downloading new mission and sample configuration parameters if available at intervals specified by **TelemetryInterval**. If Recovery mode is then deactivated, the float will begin the Park Descent phase of a new profiling cycle. If installed, a strobe light located in the helical antenna is automatically turned on at the start of Recovery mode and turned off if Recovery mode is deactivated.

The only mission.cfg file configuration parameter that applies to the Recovery mode is **ActivateRecoveryMode**:

ActivateRecoveryMode. Activates or deactivates Recovery mode.

Syntax:	ActivateRecoveryMode <on|off>
Range:	on or off
Default:	off

NOTE *By default **ActivateRecoveryMode** is off in the mission.cfg file. To place the float into recovery mode, the operator can modify the mission.cfg file to include **ActivateRecoveryMode** with no setting or with the setting set to "on." Similarly, if the float is currently in Recovery mode, sending a mission.cfg file with **ActivateRecoveryMode** set to "off" will deactivate Recovery mode.*

3.5 Emergency Mode

Emergency mode is activated only if a catastrophic error occurs during any phase or if the float fails the self test during the Mission Prelude phase. If activated, the float will immediately surface. While at the surface, the float will periodically transmit its position and log files and download new mission and sample configuration parameters if available as it does in Recovery mode. Should this situation occur, contact Teledyne Webb Research customer service. To deactivate Emergency mode, a mission.cfg file that includes an #emerg_ack entry can be downloaded. When deactivated the float will go into Recovery mode.

The only mission.cfg file configuration parameter that applies to the Emergency mode is **EmergencyTimerInterval**:

EmergencyTimerInterval. The interval in seconds between which the float attempts to communicate over the Iridium satellite network.

Syntax:	EmergencyTimerInterval <*seconds*>
Range:	5–86400 sec (24 hours)
Default:	3600 sec (1 hour)

3.6 Multimode Mission Configuration Parameters

Multimode mission configuration parameters are common to all of the four operating modes. And like the other mission configuration parameters they are located in the mission.cfg file.

The multimode mission.cfg file configuration parameters are **MinVacuum**, **VitalsMask**, **LeakDetect**, **LogVerbosity**, and **MinBuoyancyCount**.

MinVacuum. The minimum internal vacuum pressure required inside the housing. It applies only to the self test component of the Mission Prelude phase. Should the actual housing pressure exceed this value, the self test will fail.

Syntax:	MinVacuum <*decibars*>
Range:	5–11.5 dbar
Default:	9 dbar

VitalsMask. Determines the amount of vitals information that will be sent by the float in the vitals_log.bin file. For information on the vitals_log.bin file, refer to “Log Files” on page 4-17.

Syntax:	VitalsMask 0x007

LeakDetect. If set to "on," the float will transition to the Emergency mode upon detection of an internal water leak. If set to "off," the float will only log the condition to the system_log.txt file. The vitals_log.bin file provides the leak detect voltage. For information on the system_log.txt and the vitals_log.bin files, refer to “Log Files” on page 4-17.

Syntax: LeakDetect <on|off>
Range: on or off
Default: on

LogVerbosity. The verbosity level of the system_log.txt file entries transmitted over the Iridium satellite network. It corresponds to the second field of the file entries which is the priority.

Syntax: LogVerbosity <*n*>
Range: 0 (no entries transmitted) to 6 (all entries transmitted)
Default: 5

MinBuoyancyCount. The minimum allowable buoyancy position in counts, overriding any other count setting. This configuration parameter prevents the movement of oil from the oil compartment of the bladder into the oil reservoir at the specified count to manage the maximum depth of a descent. In general **MinBuoyancyCount** should be set less than the **DeepDescentCount**.

Syntax: MinBuoyancyCount <*counts*>
Range: **buoyancy_pump_min** – 2000 counts
Default: 262 counts

CAUTION *The float will not pressure activate if **MinBuoyancyCount** is greater than **MActivationCount**, as **MinBuoyancyCount** would prevent the buoyancy position from reaching **MActivationCount** which is associated with **MActivationPressure**.*

SECTION 4: Configuration and Log Files

The Teledyne Webb APEX profiling float includes two types of files: configuration files and log files. The configuration files determine the behavior of the float, its configuration and the configuration of its sensors. The log files contain the acquired data and other information gathered during the profiling cycles.

4.1 Configuration Files

The APEX profiling float includes five configuration files:

- mission.cfg
- sample.cfg
- msequence.cfg
- system.cfg
- sensors.cfg

All these files are stored on a secure digital (SD) card on the float, and each contain configuration parameters with factory default settings. For the mission.cfg, sample.cfg and msequence.cfg files only, the configuration parameter settings can be changed by the operator. The system.cfg and sensors.cfg configuration file parameters are factory set.

4.1.1 mission.cfg File

The mission.cfg file configuration parameters determine the behavior of the float during each of its operating modes as described in SECTION 3: "Operating Modes."

The following is an example of the entire contents of a mission.cfg file:

```
ActivateRecoveryMode off
AscentRate 0.08
AscentStartTimes -1 -1 -1 -1
AscentTimeout 521
AscentTimerInterval 120
BuoyancyNudge 50
DeepDescentCount 262
DeepDescentPressure 2000.00
DeepDescentTimeout 334
DeepDescentTimerInterval 300
DeepProfileFirst off
DownTime 13953
```

```
EmergencyTimerInterval 3600
HyperRetractCount 0
HyperRetractPressure 850.00
IceMonths 0000
IdleTimerInterval 3600
InitialBuoyancyNudge 300
LeakDetect on
LogVerbosity 5
MActivationCount 262
MActivationPressure 25.00
MinBuoyancyCount 262
MinVacuum 9.00
ParkBuoyancyNudge 16
ParkDeadBand 25.00
ParkDescentCount 1081
ParkDescentTimeout 334
ParkDescentTimerInterval 900
ParkPressure 1000.00
ParkTimerInterval 3600
PnPCycleLen 1
PreludeSelfTest on
PreludeTime 120
SurfacePressure 4.00
TelemetryDays 0 0
TelemetryInterval 900
TelemetryTimeout 120
UpTime 641
VitalsMask 0007
```

These configuration parameters can be changed by the operator, either locally by using a PC connection to the float, or remotely between missions by using the Iridium satellite network. For instructions on how to change the mission.cfg file configuration parameters, refer to SECTION 5: "Modifying the Mission Plan."

4.1.2 sample.cfg File

The sample.cfg file configuration parameters allow the operator to program the sampling behavior of the float by enabling interval sampling of the CTD and the optional sensors as well as continuous profiling with the CTD over one or more ranges of pressure during one or more phases in Mission mode. For information on configuring the sampling of optional sensors, refer to APPENDIX A: "Optional Sensors."

Interval sampling and continuous profiling can be performed separately or simultaneously. The sample.cfg file parameters can be changed by the operator remotely between missions by using the Iridium satellite network. For instructions on how to change the sample.cfg file parameters, refer to “Modifying the sample.cfg File Configuration Parameter Settings Remotely” on page 5-4, and for more information on the sample.cfg file along with examples for the specific CTD installed on the float, refer to SECTION 6: “CTDs.”

Interval Sampling. Interval sampling specifies, for the CTD and any one or more of the optional sensors, the pressure at which sampling will begin, the pressure at which sampling will end, and the pressure or time intervals at which a sample will occur. This sampling behavior is specified by entering the phase followed by a single line entry in the sample.cfg file for *each* of any number of sensor and pressure range combinations. When interval sampling two or more sensors at the same depth, they are sampled in the order specified in the sample.cfg file. Interval sampling can also be performed during the Park and Surface phases by specifying a time interval. For the Park phase, the interval units should be time specified. If the interval is set to 0, **ParkTimerInterval** is used to determine the sampling interval.

Syntax: *<phase>*
SAMPLE *<sensor type> [<start> [<stop> [<interval> [<units> [<count>]]]]]*

phase:
PARKDESCENT
PARK
DEEPDESCENT
ASCENT
SURFACE

sensor type:
CTD - conductivity, temperature and pressure sensor
LGR - Multichannel CTD
OPT - oxygen sensor
FLBB - fluorometer
XMIS - transmissometer
RAD - radiance radiometer
IRAD - irradiance radiometer
CMP - compass
SUNA - nitrate sensor

start: Start pressure (0–2500) dbar

stop: Stop pressure (0–2500) dbar

interval: Pressure interval (0-100) dbar

units: DBAR or SEC

count: Number of samples

NOTE *The float does not descend to 2500 dbar. The configured maximum value is used so that the float will immediately start profiling at depth when it begins the Ascent phase.*

Brackets ([ ]) around a parameter or group of parameters indicate that it is optional, and entries are *not* case sensitive. The brackets, however, should *not* be included in the entry. In addition, sampling during the Surface phase can *only* be performed for sensors that are capable of sampling at the surface.

For a sample.cfg file interval sampling entry that *does not* include one or more of the optional parameters, the float uses the following default entries for the specified sensor:

start:	2500
stop:	0
interval:	0
units:	DBAR
count:	1

An interval parameter of 0 is a special case where the timer interval specified for the mission phase, for example, **AscentTimerInterval** is used to determine the sampling interval for time-based rather than pressure-based interval sampling. Therefore, to sample based on time intervals in a mission phase instead of pressure, set interval to 0 and make the required interval time setting for the associated timer interval for the mission phase.

The start and stop pressures can be in any order. Equal start and stop pressure entries with *no* entry for the interval parameter causes the sensor to be sampled once at the specified pressure. Interval sampling of a pumped type CTD cannot be performed at the surface. For the Park phase, **ParkTimerInterval** is used to determine the sampling interval. Refer to “ParkTimerInterval” on page 3-9 for information on **ParkTimerInterval**. For this phase the start, stop and interval parameters do not apply.

NOTE *For the best accuracy, it is recommended that interval sampling between 750 meters and the surface be performed simultaneously with continuous profiling, except when **tswait**=20 or greater. Refer to SECTION 6: "CTDs.”*

Example 1. To sample the CTD during the Ascent phase once at 2000 dbar, once at 1000 dbar, and then every 10 dbar beginning at 400 dbar and ending at 50 dbar, enter:

```
<ASCENT>
SAMPLE CTD 2000 2000 DBAR
SAMPLE CTD 1000 1000 DBAR
SAMPLE CTD 400 50 10 DBAR
```

Example 2. To sample the CTD PTS every 100 dbar beginning at 2000 dbar and ending at 1000 dbar, then again every 50 dbar ending at 50 dbar; the oxygen sensor at 2000 dbar and every 100 dbar beginning at 1000 dbar and ending at 100 dbar; and the fluorometer at 2000 dbar and every 100 dbar beginning at 1500 dbar and ending at 100 dbar, enter:

```
<ASCENT>
SAMPLE PTS 2000 1000 100 DBAR
SAMPLE PTS 950 50 50 DBAR
SAMPLE OPT 2000 2000 DBAR
SAMPLE OPT 1000 100 100 DBAR
SAMPLE FLBB 2000 2000 DBAR
SAMPLE FLBB 1500 100 100 DBAR
```

For a graphical representation of this sampling behavior, refer to Figure 4-1.

Example 3. To sample the CTD PTS, the compass and the radiance and irradiance radiometers, in that order, during the entire ascent and at the intervals specified by **AscentTimerInterval**, enter:

```
<ASCENT>
SAMPLE PTS 2000 4 0 DBAR
SAMPLE CMP 2000 4 0 DBAR
SAMPLE RAD 2000 0 0 DBAR
SAMPLE IRAD 2000 0 0 DBAR
```

or

```
<ASCENT>
SAMPLE PTS
SAMPLE CMP
SAMPLE RAD
SAMPLE IRAD
```

Example 4. To sample the CTD PTS of a float with a pumped type CTD during the Park phase at an interval of every 8 hours enter:

<PARK>
SAMPLE PTS 2500 4 28800 SEC

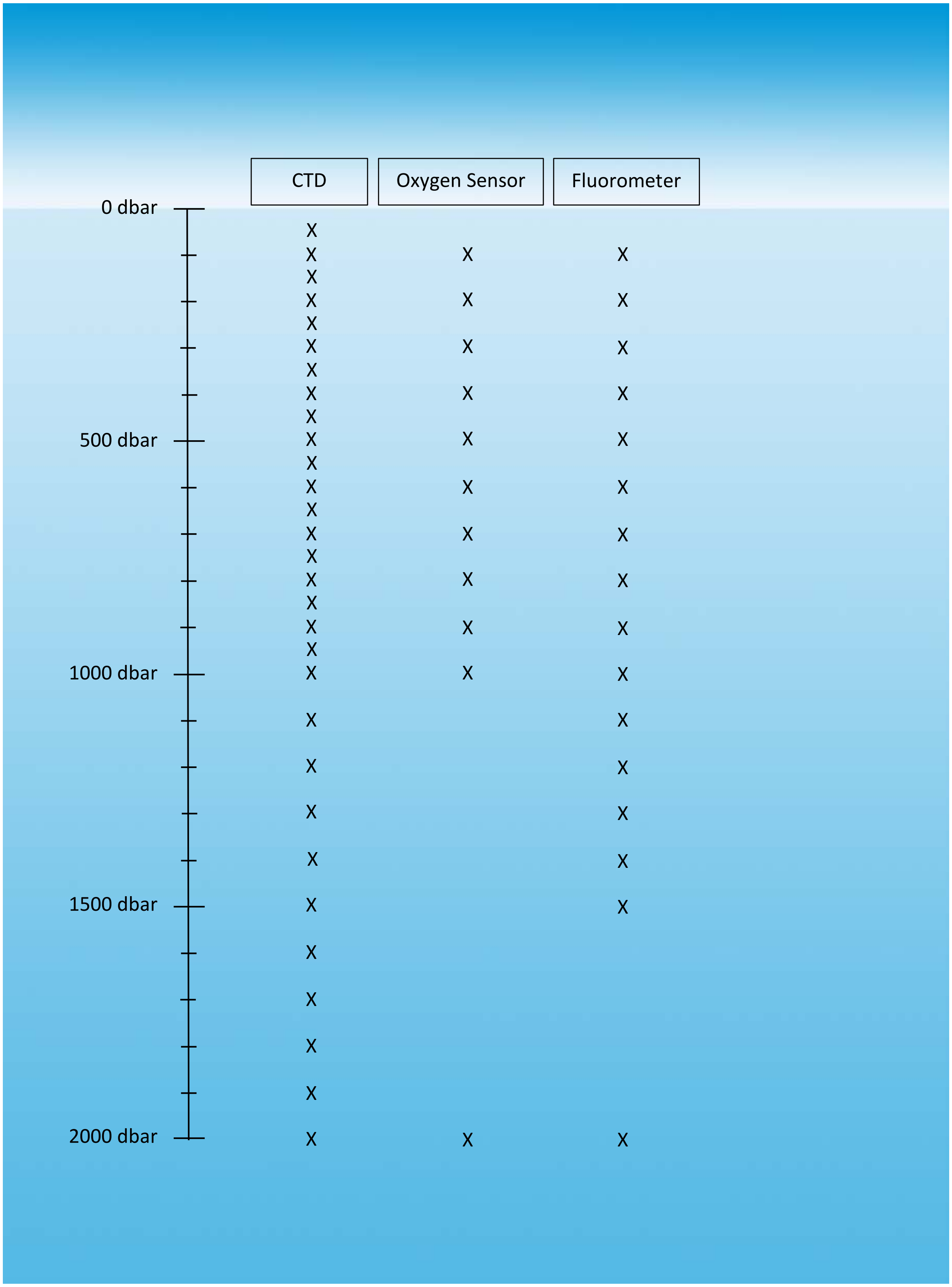


Figure 4-1: Example Sampling Behavior from a sample.cfg File

Example 5. To sample the CTD PTS of a float with a non-pumped type CTD during the Surface phase, enter:

```
<SURFACE>
SAMPLE LGR
```

Example 6. To sample the CTD PTS 10 times at 500 dbar, enter:

```
<ASCENT>
SAMPLE PTS 500 500 DBAR 10
```

Example 7. To sample the CTD PTS 5 times each at 1500 dbar, 1000 dbar and 500 dbar, enter:

```
<ASCENT>
SAMPLE PTS 1500 1500 DBAR 5
SAMPLE PTS 1000 1000 DBAR 5
SAMPLE PTS 500 500 DBAR 5
```

Example 8. To sample the CTD PTS 5 times every 100 dbar beginning at 1000 dbar and ending at 500 dbar, enter:

```
<ASCENT>
SAMPLE PTS 1000 500 100 DBAR 5
```

Continuous profiling. Continuous profiling, which applies to the CTD only, specifies for the Ascent or Descent phases the pressure at which continuous profiling will begin and the pressure at which continuous profiling will end. For a non-pumped type CTD, both the sample rate in samples per second and a bin size can also be specified, where the bin size is the pressure in decibars over which averaging is performed. For a pumped type CTD, the bin size is fixed at 2 decibars, and the sample rate is fixed at 1 sample per second. This sampling behavior is specified for the Ascent phase by entering the phase followed by a single entry in the sample.cfg file for each of any number of pressure ranges.

Syntax: *<phase>*
PROFILE *<sensor type> [<start> [<stop> [<bin_size> [<rate>]]]]*

phase: PARKDESCENT
DEEPDESCENT
ASCENT

sensor type: CTD - conductivity, temperature and pressure sensor
LGR – non-pumped type CTD

start: Start pressure (0–2100) DBAR

stop: Stop pressure (0–2100) DBAR

CTD:

bin size:	2 DBAR
rate:	1 HZ

LGR:

bin size:	1–10 DBAR 0 DBAR for raw data
rate:	1, 2, 4 or 8 HZ 1–60 SEC per sample

Brackets ([ ]) around a parameter or group of parameters indicate that it is optional, and entries are *not* case sensitive. The brackets, however, should *not* be included in the entry.

CAUTION *A bin size of less than 1or a rate of more than 1, or both, can result in excessively large data files and battery usage. For dense sampling over one or more specific, relatively short pressure ranges, contact Teledyne Webb Research customer service for instructions.*

For RBR*argo*^{3} C.T.D rate entries in samples/second and their corresponding sample intervals in seconds, refer to Table A-1. The default is to sample at 1 sample/second (1 Hz).

Table A-1: RBR*argo*^{3} C.T.D Rate Entries Versus Sample Interval

CTD Sample Interval Rate Entry (samples/second)	Per Sample Interval (seconds)
8	.125
4	.25
2	.50
1	1

For a sample.cfg file continuous profiling entry that does not include one or more of the optional parameters, a float with a *pumped* type CTD uses the following default entries:

start:	2000
stop:	**SurfacePressure** (Refer to “SurfacePressure” on page 3-16.)
bin_size:	2 DBAR
rate:	1 HZ

For a stop of less than **SurfacePressure**, continuous profiling will stop at the "pcutoff" pressure of the CTD. For a *non-pumped* type CTD, the default settings are the following:

start:	2000
stop:	0 (Stops when conductivity is 0 in air)
bin_size:	2 DBAR
rate:	1 HZ

The entry or series of entries in the sample.cfg file must be preceded with a single line to indicate sampling during the Ascent phase as follows:

<ASCENT>

<PARKDESCENT>

<DEEPDESCENT>

The start and stop pressures can be in any order, and continuous profiling with the CTD all the way to the surface can *only* be performed for a float with a non-pumped type CTD.

Example 1. To sample a pumped type CTD continuously beginning at 2000 dbar and ending at 1500 dbar, enter:

```
<ASCENT>
PROFILE CTD 2000 1500
```

Example 2. To sample a pumped type CTD continuously beginning at 2000 dbar and ending at the pressure specified by **SurfacePressure**:

```
<ASCENT>
PROFILE CTD 2000 0
```

or

```
<ASCENT>
PROFILE CTD
```

Example 3. To sample a non-pumped type CTD continuously beginning at 2000 dbar and ending at 1500 dbar with a bin size of 5 and a sample rate of 1, enter:

```
<ASCENT>
Profile CTD 2000 1500 5 1 HZ
```

Example 4. To sample a non-pumped type CTD continuously beginning at 1000 dbar for the entire remaining ascent with a bin size of 2 and a sample rate of 1, enter:

```
<ASCENT>
PROFILE LGR 1000 0 2 1 HZ
```

Example 5. To sample a non-pumped type CTD continuously beginning at 2000 dbar and ending at 1500 dbar with a bin size of 8 and a sample rate of 1 HZ, and then again beginning at 1500 dbar and ending at 1000 dbar with a bin size of 4 and a sample rate of 2 HZ, and then for the entire remaining ascent with a bin size of 2 and a sample rate of every 5 seconds, enter:

```
<ASCENT>
PROFILE LGR 2000 1500 8 1 HZ
PROFILE LGR 1500 1000 4 2 HZ
PROFILE LGR 1000 0 2 5 SEC
```

Example 6. To sample a non-pumped type CTD every 50 dbar beginning at 2000 dbar and ending at 1500 dbar, and then every 20 dbar beginning at 1500 dbar and ending at 1000 dbar, and then continuously beginning at 1000 dbar for the entire remaining ascent with a bin size of 5 and a sample rate of 1, enter:

```
<ASCENT>
SAMPLE LGR 2000 1500 50 DBAR
SAMPLE LGR 1500 1000 20 DBAR
PROFILE LGR 1000 0 5 1 HZ
```

Example 7. To sample a pumped type CTD every 50 dbar beginning at 2000 bar and ending at 1500 dbar, and then every 20 dbar beginning at 1500 dbar and ending at 1000 dbar, while performing continuous profiling beginning at 2000 dbar for the entire ascent, enter:

```
<ASCENT>
SAMPLE CTD 2000 1500 50 DBAR
SAMPLE CTD 1500 1000 20 DBAR
PROFILE CTD 2000 0
```

or

```
<ASCENT>
SAMPLE CTD 2000 1500 50 DBAR
SAMPLE CTD 1500 1000 20 DBAR
PROFILE CTD
```

In addition to the stop parameter, other conditions will stop continuous profiling:

- Continuous profiling will stop at the end of the period specified by **AscentTimeout**.
- For a float with a pumped type CTD, continuous profiling will stop when it reaches the pressure specified by **SurfacePressure** or when the "pcutoff" pressure of the CTD is reached, whichever occurs first.
- For a float with a non-pumped type CTD, continuous profiling will stop when it breaks the surface, which is when zero conductivity is measured.

- Continuous profiling will stop should Emergency mode be activated.
- Continuous profiling will stop upon detection of ice if ice avoidance is enabled.

4.1.3 msequence.cfg File

The msequence.cfg file configuration parameters enable up to five missions to be stored on the float. When specified, the float will automatically update the mission.cfg file configuration parameters when the previous set completes. The changes are applied when the float is in the Surface phase prior to starting the next mission cycle as if new mission.cfg and sample.cfg files were downloaded during telemetry. When the float completes the last mission within a sequence, the float will updates its mission parameters to include the mission parameters specified in the first mission sequence. Each mission in the mission sequence builds off the prior mission. During the mission cycles, the **ParkDescentCount** and **DeepDescentCount** configuration parameters are dynamically modified by the float as it learns the count associated with the specified depth. Unless overwritten by a new parameter value, the learned counts are preserved as the float updates the new mission parameters. The new set of parameters specified are updated to the current set of active parameters. For example, if only two parameters are specified, those parameters will be updated and the remaining active parameters will retain their current values. If the new set of specified parameters do not pass verification, the float will log and error message and continue to use the current set of parameters. The msequence.cfg file configuration parameters can be changed by the operator remotely between missions by using the Iridium satellite network. For instructions on how to change the msequence.cfg file configuration parameters, refer to SECTION 5: “Modifying the Mission Plan.”

The mission sequence configuration specification is divided into 3 main sections.

Sequence section. The first section [SEQ] allows the user to specify an ordered list of up to 5 different missions to run in sequence. The mission identifiers are M1, M2, M3, M4 and M5. For each mission identifier, the number of cycles to run must also be specified. The minimum number of cycles for a mission is 1. The maximum number of cycles to run is 255. For example, “M1 20” specifies to run mission M1 for 20 mission cycles. The mission parameters and sample configuration to apply to the M1 mission are specified in later sections. The [SEQ] section is optional for msequence.cfg file updates as long as a prior msequence.cfg file included a valid [SEQ] section. When omitted, the float will sequence in the order of the missions specified in the last known valid msequence.cfg file.

Mission parameter section. Following the sequence section [SEQ] the user specifies the mission parameter configurations for the missions that are defined in the sequence section. For example, if the [SEQ] section specifies missions M1, M2 and

M3, one can specify mission parameter sections [M1], [M2] and [M3]. The mission configuration sections are optional, however the expectation is that they will specify the mission parameters to use in the sequenced missions. Each mission parameter section only needs to specify the modified mission parameters similar to a new mission.cfg file. Each new mission configuration in the mission sequence builds off the prior mission configuration. For example, if the [SEQ] section specifies M1, M2 and then M3. The mission configuration for M1 builds off the current mission configuration. The mission configuration for M2 builds off the M1 mission configuration. The mission configuration for M3 builds off the M2 configuration. The next time mission M1 is run in the sequence, it builds off the mission configuration for M3. If a mission configuration section for a mission is not specified, the mission configuration for the prior mission will be used.

Sample parameter section. Following the mission configuration sections, the user specifies the sample configuration sections for each mission in the sequence. For example, if the [SEQ] section specifies missions M1, M2 and M3, one can specify sections [S1], [S2] and [S3]. The sample configuration sections are not required, however it may be desirable to perform different sampling behavior for each mission. The sample configuration sections must provide the full sample configuration for each mission. If a sample configuration section for a mission is not specified, the sample configuration from the prior mission will be used.

The msequence.cfg file configuration sections are [SEQ], [M1], [M2], [M3], [M4], [M5], [S1], [S2], [S3], [S4] and [S5]. The APF11 firmware will add a CHECKSUM section to the bottom of an msequence.cfg file upon being parsed. The msequence.cfg syntax is below.

Syntax: [SEQ]
M<*n*> *<count>*
[M<n>]
<parameter> <value>
[S<n>]
<phase>
SAMPLE *<sensor type> [<start> [<stop> [<interval> [<units> [<count>]]]]]*
PROFILE *<sensor type> [<start> [<stop> [<bin_size> [<rate>]]]]*

n:	Mission sequence number (1 - 5)
parameter	Mission parameter
value:	Mission parameter value
phase:	Mission phase (PARKDESCENT, PARK DEEPDESCENT, ASCENT, SURFACE)

For the list of configuration parameters available, refer to "Mission Mode" on page 3-4. For information about configuring the sampling behavior, refer to "sample.cfg File" on page 4-2.

The following is an example of the entire contents of an msequence.cfg file where the float runs an initial shorter 2 day mission to 1000 meters for 4 cycles and then runs 10 day cycles for 250 cycles before activating Recovery Mode. For the first 4 cycles the float samples PTS during the Park phase every ParkTimerInterval before configuring sampling PTS at Park every 4 hours:

```
[SEQ]
M1 4
M2 250
M3 1
[M1]
ActivateRecoveryMode off
AscentTimeout 260
DeepDescentCount 1081
DeepDescentTimeout 175
DeepDescentPressure 1000
DownTime 2710
ParkDescentCount 1371
ParkDescentTimeout 175
ParkPressure 500
UpTime 321
[M2]
AscentTimeout 521
DeepDescentCount 262
DeepDescentPressure 2000
DeepDescentTimeout 334
DownTime 13953
ParkDescentCount 1081
ParkDescentTimeout 334
ParkPressure 1000
UpTime 641
[M3]
ActivateRecoveryMode on
[S1]
<PARK>
SAMPLE OPT 2500 0 0 DBAR 1
SAMPLE FLBB 2500 0 0 DBAR 1
SAMPLE PTS 2500 4 0 DBAR 1
<ASCENT>
SAMPLE OPT 250 0 2 DBAR 1
SAMPLE FLBB 250 0 2 DBAR 1
SAMPLE PTS 250 4 2 DBAR 1
```

```
PROFILE CTD 250 4 2 1 HZ #Regime:1
<SURFACE>
MEASURE OPT
[S2]
<PARK>
SAMPLE OPT 2500 0 0 DBAR 1
SAMPLE FLBB 2500 0 0 DBAR 1
SAMPLE PTS 2500 4 240 SEC 1
<ASCENT>
SAMPLE OPT 250 0 2 DBAR 1
SAMPLE FLBB 250 0 2 DBAR 1
SAMPLE PTS 250 4 5 DBAR 1
PROFILE CTD 2500 4 2 1 HZ #Regime:1
<SURFACE>
MEASURE OPT
```

For this example, the M1 mission configuration parameters specified are used for 4 mission cycles with the specified sampling configuration. After that cycle, the M2 mission parameters are applied for 250 cycles with the sample configuration changes for the [S2] parameters. At the end of the 254th the float goes into recovery mode at the surface. This pattern applies until modified by the user by sending a new msequence.cfg file to the float. A new msequence.cfg file can be provided to the float to change the mission and sample configuration similar to using updated mission.cfg and sample.cfg files. For a standard Iridium float, providing a mission sequence update is performed by placing a new msequence.cfg file in the float's home directory. When the float connects to server and performs a successful login, the APF11 will first look for an msequence.cfg file. If one does not exist, the float will look for a mission.cfg and sample.cfg. If the APF11 finds a mission.cfg or sample.cfg file in the float's home directory, that will result in cancellation of mission sequencing. For an SBD Iridium float, providing a mission sequence update is performed by emailing a new msequence.cfg file to the float. When the float connects via SBD, it will download files in the order they were sent. This is important because if one sends a new msequence.cfg file and then sends a mission.cfg or sample.cfg, when the APF11 downloads a mission.cfg or sample.cfg, that will result in cancellation of mission sequencing.

4.1.4 system.cfg File

The system.cfg file configuration parameters set the maximum air bladder inflation, the minimum and maximum positions for the buoyancy pump and some Iridium satellite network values. These configuration parameter settings are factory set and should not be changed by the operator. The factory settings, as listed below, may also be different from float to float.

The system.cfg file configuration parameters are the following:

air_bladder_max. The maximum pressure in decibars to which the air bladder will inflate. The air pump will shut off if the air bladder pressure reaches this value.

Syntax:	air_bladder_max <*dbar*>
Factory setting:	12 dbar

buoyancy_pump_min. The minimum buoyancy position in counts. The buoyancy pump will shut off if the buoyancy position reaches this minimum value.

Syntax:	buoyancy_pump_min <*counts*>
Default:	147 counts (typical, may vary depending on the amount of oil)

buoyancy_pump_max. The maximum buoyancy position in counts. The buoyancy pump will shut off if the buoyancy position reaches this maximum value.

Syntax:	buoyancy_pump_max <*counts*>
Default:	3720 counts (typical, may vary depending on the amount of oil)

float_id. The Controller board serial number specified in NNN-WWYY format, where NNN is a number from 001 to 999, WW is a week from 01 to 52, and YY is a year from 00 to 99.

Syntax:	float_id <*float id*>

iridium. For Iridium RUDICS/PSTN the system.cfg file also includes account information, which is the user name, the login password and the primary and alternate dial string parameters required for connecting to a server over the Iridium satellite communications network. For SBD the system.cfg file *does not* include account information.

Syntax:	iridium <*username*> <*password*> <*primary_dial_string*> dialup |rudics <*alternate_dial_string*> dialup | rudics
Example:	iridium apf11 apex 00881600005135 rudics 0015085480980 dialup

The primary dial string is used by the float first. The alternate dial string is used only if the primary dial string fails to connect after a fixed number of tries. The dialup option enables the float to connect to a server using the Iridium satellite network, the Public Switched Telephone Network (PSTN) and a modem. The RUDICS option enables the float to connect to a server over the internet using the Iridium satellite network and Router-Based Unrestricted Digital Internetworking Connectivity Solution (RUDICS). Refer to “Setting up RUDICS/PSTN Communications with the Float” on page F-1 for information on how to set up to connect to the float using these systems.

The following is an example of the entire contents of a system.cfg file:

```
float_id 071-4517
iridium apf11 apex 00881600005135 rudics 0015085480980 dialup
air_bladder_max 12
buoyancy_pump_min 147
buoyancy_pump_max 3720
```

In this example RUDICS is used for the primary dial string, and dialup for the alternate.

4.1.5 sensors.cfg File

The sensors.cfg file configuration parameters define the CTD hardware configuration along with the GPS receiver, the Iridium modem and the optional sensors. The sensors.cfg file configuration parameters should not be changed by the operator.

The sensors.cfg file configuration parameters are **sensor**, **gps**, **modem**, and optional **strobe**.

sensor. Specifies the name of the CTD and its hardware configuration.

Syntax: *<sensor_type>* *<com_port>* *<baud_rate>*

Default: *sensor type*: seabird_ctd_41cp (pumped type CTD)
com_port: COM2
baud_rate: B9600

Optional: *sensor type*: seabird_ctd_41 (pumped type CTD)
seabird_ctd_41n (pumped type CTD)
rbr_logger (non-pumped type CTD)

Example: seabird_ctd_41cp COM2 B9600

gps. Specifies the name of the GPS receiver. The name is used to define its internal hardware configuration.

Syntax: gps *<gps_name>*

Default: gps15xh

Example: gps gps15xh

modem. Specifies the name of the Iridium modem. The name is used to define its internal hardware configuration.

Syntax: modem *<modem_name>*

Default: iridium9523

Optional: sbd9602

Example: modem iridium9523

The following is an example of the entire contents of a sensors.cfg file:

```
seabird_ctd_41cp COM2 B9600
gps gps15xl
modem iridium9523
```

For the sensor hardware configuration for the optional sensors, refer to APPENDIX A: “Optional Sensors.”

4.2 Log Files

The APEX profiling float includes up to four log files. These files are transferred *from* the float *to* the server and are referred to throughout this manual as the following:

- science_log.bin
- vitals_log.bin
- system_log.txt
- critical_log.txt (Provided only for unusual, exceptional error conditions.)

The *actual* file names are prepended with the username, the profile number, the timestamp, and the filetype, and they are appended with an extension as follows:

Syntax:	*< username>.<profile number>.<timestamp>.<filename>.<filetype>.<ext>*
username:	The account username included in system.cfg.
profile number:	A zero-padded 3-decimal number indicating the profile number. For example, the first descent would have a value of 001.
timestamp:	Date and time of the file creation in YYYYMMDDTHHMMSS format
filename:	Name of file
filetype:	Type of log file (.bin or .txt)
ext:	gz

NOTE *<username> is used for RUDICS/PSTN communications only. For SBD and Argos communications, <float id> is used instead of <username>.*

For example, in the following files the float id is 071-4517 and the profile number is 003:

- **071-4517.003.20160122T193806.**science_log.bin**.gz**
- **071-4517.003.20160122T193806.**vitals_log.bin**.gz**
- **071-4517.003.20160122T193806.**system_log.txt**.gz**
- **071-4517.003.20160122T193806.**critical_log.txt**.gz**

The data in the science_log.bin and vitals_log.bin files are in compressed binary format. These two files must be unzipped into uncompressed binary files and then converted to comma separated value (CSV) files that can be imported into a spreadsheet program. The system_log.txt and critical_log.txt files are text files containing ASCII characters and can be opened in any text editor, such as Microsoft Notepad. All the files are transmitted over the Iridium satellite network prior to the start of each profiling cycle.

4.2.1 science_log.bin File

The science_log.bin file stores profile data, float position and other information:

- Firmware version, username and float id
- Sensor type
- Time stamp
- Pressure
- Temperature
- Salinity
- Surface position
- pH

Contact Teledyne Webb Research customer service for information on .bin file formats.

4.2.2 vitals_log.bin File

The vitals_log.bin file stores time stamped system status data:

- Firmware version and float id
- Time stamp
- Air bladder pressure
- Air bladder pressure in counts
- Battery voltage
- Battery voltage in counts
- Humidity
- Leak detect voltage
- Internal vacuum pressure
- Internal vacuum pressure in counts
- Coulomb counter
- Battery current draw
- Battery current raw
- RSSI value, where RSSI is the Iridium signal strength

4.2.3 system_log.txt File

The system_log.txt files stores system events, verbosities, errors, messages, and other information in the following format:

<time stamp>|<priority>|<function_name>|<message>

4.3 Converting the science_log.bin and vitals_log.bin Files to CSV Files

The science_log.bin and vitals_log.bin files must be uncompressed and then converted to CSV files before they can be opened in a spreadsheet program. This operation requires that you first download and install Python. To install Python, go to http://www.python.org/getit/releases/2.7.11 and download and install Version 2.7.11 in the C:\ directory of your server. In addition, the apf11dec.py script, provided by Teledyne Webb Research, must also be saved in the C:\ directory of the server. This script is on the documentation CD.

To convert the science_log.bin and vitals_log.bin files to CSV files:

1. Locate the science_log.bin.gz or vitals_log.bin.gz file in the home directory of the server. The actual file names will additionally include the float name, the profile number and a timestamp as described in “Log Files” on page 4-17.
2. Enter `gunzip <filename>`.

   The specified file will be unzipped to the corresponding science_log.bin or vitals_log.bin file with no .gz extension.
3. Enter `python apf11dec.py <filename>`.

   The specified binary science_log.bin or vitals_log.bin file will be used to generate a corresponding science_log.csv or vitals_log.csv file.
4. Open the CSV file in a spreadsheet program, such as Microsoft Excel.

4.4 Reading the science_log.csv File

An example printout of a science_log.csv file with CTD data is shown in Figure 4-2. In this example only several rows of data are shown to illustrate the structure of the log file. Typically there would be many rows of data.

The columns for the Message rows are from left to right:

- Time stamp in UTC (YYYYMMDDTHHMMSS)
- Firmware version or float id

Message	20161121T223335	Firmware: 07/18/16 20:08:07 DEEP_APEX-v2.3.18			
Message	20161121T223335	FloatId/Username: f0001			
GPS	20161121T223335	-58.6827	-56.0524	5	
CTD_P	20161121T223357	-0.29			
Message	20161121T223357	Park Descent Phase*********			
CTD_P	20161121T223412	-0.3			
CTD_P	20161121T223416	-0.4			
CTD_P	20161121T223515	-0.38			
• • •					
CTD_P	20161122T135830	1484.48			
Message	20161122T135835	Park Phase*****************			
CTD_P	20161122T140340	1483.73			
Message	20161122T140340	Deep Descent Phase***************			
CTD_P	20161122T140447	1483.62			
Message	20161122T140452	Profiling Phase************			
CTD_P	20161122T140455	1483.72			
Message	20161122T140502	Continuous Profile Started**********			
CTD_PTS	20161122T140526	1483.47	1.3397	34.707	
CTD_PTS	20161122T140741	1476.69	1.3475	34.7078	
CTD_PTS	20161122T141248	1448.63	1.3648	34.708	
• • •					
CTD_PTS	20161122T174039	1.94	0.2297	33.6195	
Message	20161122T174047	Surface Phase**************			
CTD_bins	20161122T174441	12917	743	1483.59	
CTD_CP	20161122T174832	2.27	0.2294	33.6214	11
CTD_CP	20161122T174832	3.98	0.2295	33.6217	15
CTD_CP	20161122T174832	6	0.2377	33.6214	15
• • •					
CTD_CP	20161122T174911	1483.36	1.3385	34.7081	41
CTD_P	20161122T180349	0.04			
GPS	20150214T033525	-58.7147	-55.9328	9	

Figure 4-2: Example science_log.csv Printout

The columns for the GPS rows are from left to right:

- Time stamp in UTC (YYYYMMDDTHHMMSS)
- Latitude in decimal degrees (±DD.DDDDD) of the surface position.
- Longitude in decimal degrees (±DDD.DDDDD) of the surface position
- Number of GPS satellites used

The columns for the CTD_P rows under Park Descent Phase, Deep Descent Phase, Profiling Phase, and Surface Phase contain CTD pressure data that were recorded when interval sampling and are from left to right:

- Time stamp in UTC (YYYYMMDDTHHMMSS)
- Pressure in decibars

NOTE *If optional sensors are installed, a single CTD_P row will be automatically added before and after a series of CTD and optional sensor samples.*

The columns for the CTD_PTS rows under Profiling Phase contain sensor data that were recorded when interval sampling and are from left to right:

- Time stamp in UTC (YYYYMMDDTHHMMSS)
- Pressure in decibars
- Temperature in degrees Celsius
- Salinity in PSU

Although not shown in the example printout, columns under Profiling Phase that contain *only* pressure and temperature data that were recorded when interval sampling would be associated with CTD_PT rows and would be from left to right:

- Time stamp in UTC (YYYYMMDDTHHMMSS)
- Pressure in decibars
- Temperature in degrees Celsius

The columns for the CTD_bins row under Profiling Phase contain bin averaging statistics which apply *only* to a float with a pumped type CTD and are from left to right:

- Time stamp in UTC (YYYYMMDDTHHMMSS)
- Number of samples recorded during the mission
- Number of bins recorded during the mission
- Highest pressure in decibars recorded during the mission

The columns for the CTD_CP rows under Profiling Phase contain sensor data for a float with a pumped type CTD and were recorded when continuous profiling. Similarly, for a float with a non-pumped type CTD, the columns are associated instead with CTD_CP+ rows under Profiling Phase. For this float there is one additional data column which contains conductivity data.

NOTE *When continuous profiling with a float that uses a pumped type CTD, the time stamps in the science_log.bin file are* <u>*not*</u> *the times at which the samples were actually taken. Rather the time stamps are the times at which the data were recorded to the science_log.bin file which occurs when continuous profiling stops. When continuous profiling with a float that uses a non-pumped type CTD, the time stamps are provided by the CTD, and they are the times at which the samples were taken at depth.*

For a float with a pumped type CTD, the columns are from left to right:

- Time stamp in UTC (YYYYMMDDTHHMMSS)
- Bin averaged pressure in decibars
- Bin averaged temperature in degrees Celsius
- Bin averaged salinity in PSU
- Number of samples averaged per bin

For a float with a non-pumped type CTD, the columns are from left to right:

- Time stamp in UTC (YYYYMMDDTHHMMSS)
- Bin averaged pressure in decibars
- Bin averaged temperature in degrees Celsius
- Bin averaged salinity in PSU
- Conductivity in mS/cm
- Number of samples averaged per bin

NOTE *If optional sensors are installed, additional rows will be included in the science_log.csv printout. For information on the fields in these rows, refer to APPENDIX A: "Optional Sensors."*

4.5 Reading the vitals_log.csv File

An example printout of a vitals_log.csv file is shown in Figure 4-3 . Typically this file would contain many more rows of data, but for illustration purposes, only several are shown. The vitals_log.csv file is updated at the start of each mission phase.

The columns for the Message rows are from left to right:

- Time stamp in UTC (YYYYMMDDTHHMMSS)
- Firmware version, username and float id

The columns for the VITALS_CORE rows are from left to right:

- Time stamp in UTC (YYYYMMDDTHHMMSS)
- Air bladder pressure in decibars
- Air bladder pressure in counts
- Battery voltage in volts
- Battery voltage in counts
- Humidity in percent relative
- Leak detect voltage in volts
- Internal vacuum pressure in decibars
- Internal vacuum pressure in counts
- Coulomb counter in milliampere hours
- Battery current draw in milliamperes
- Battery current raw in counts

The columns for the RSSI rows are from left to right:

- Time stamp in UTC (YYYYMMDDTHHMMSS)
- RSSI value, where RSSI is the Iridium signal strength from 1 to 5 where 5 is the best

The vitals parameters are the following:

Air Bladder Pressure. The float reports the internal air bladder pressure in decibars and corresponding counts. When the air bladder is deflated, it is expected that the air pressure will measure approximately the same as the internal vacuum of about 7.8 decibars. When the air bladder is inflated it would measure closer to 12 decibars.

	Time Stamp (UTC)	Air Bladder Pressure (dbar)	Air Bladder Pressure (counts)	Battery Voltage (volts)	Battery Voltage (counts)	Humidity (%)	Leak Detect Voltage (volts)	Internal Vacuum Pressure (dbar)	Internal Vacuum Pressure (counts)	Coulomb Counter (mAh)	Battery Current Draw (mA)	Battery Current Draw (counts)
Message	20170524T054717	Firmware: 03/01/19 21:58:07 2.12.3 STD Final										
Message	20170524T054717	Username: apf11										
Message	20170524T054717	Float ID: 014-2019										
VITALS_CORE	20170524T074714	10.01	2026	13.69	3115	50	2.3	10.40	4086	1.56	32.5	-421
RSSI	20170524T075218	4										
RSSI	20170524T083622	5										
RSSI	20170524T085519	5										
VITALS_CORE	20170524T095101	11.97	2412	13.69	3117	49	2.3	10.41	4087	7.81	0.0	0
VITALS_CORE	20170524T100035	10.00	2026	13.69	3115	48	2.3	10.40	4086	27.19	0.0	0
VITALS_CORE	20170524T100805	10.00	2024	13.68	3115	50	2.3	10.4	4082	29.70	69.2	-887
VITALS_CORE	20170524T100814	9.99	2024	13.68	3114	50	2.3	10.39	4083	29.27	0.0	0
VITALS_CORE	20170524T100956	10.00	2025	13.69	3117	50	2.3	10.39	4085	30.95	0.0	0
VITALS_CORE	20170524T101452	10.00	2025	13.69	3116	49	2.3	10.40	4085	34.75	0.0	0

Figure 4-3: Example vitals_log Printout

Battery Voltage. The float reports the measured voltage and corresponding counts of the internal batteries. For Alkaline batteries, this is approximately 15 volts and decreases over time as the float runs profiling cycles. For Lithium batteries the voltage is more than 15 volts and remains the same for most of profiling cycles until a rapid decrease as the batteries run out of energy. This can be used to determine when the battery energy is nearing depletion.

Humidity. The float reports its internal humidity which is expected to be less than 10%. If the internal humidity starts to increase, it could be a sign that seawater is getting into the float.

Leak Detect Voltage. When equipped with a leak detect sensor, the float measures the leak detect voltage which is expected to be about 2.3 volts. If seawater gets into the float the leak detect voltage will drop to 0 and the float will report the event.

Internal Vacuum Pressure. The float reports its internal vacuum which is set to less than 8.0 decibars. An increase could indicate that the float is not holding its vacuum.

Coulomb Counter. The float measures the total amount of milliamperes used to date as it runs each mission cycle and takes a reading at key events of a mission cycle. This can be used to approximate the number of cycles expected based on the configured mission parameters and sampling behavior.

Battery Current Draw. The float measures the current draw in milliamperes on the batteries. This can be used to determine of the hardware devices are properly operating.

4.6 Converting system_log.txt Files to Readable Text

To convert the system_log.txt file to readable text, enter `gunzip system_log.txt.gz`. The file will be unzipped to the system_log.txt file which can be opened in a text editor.

The following is an example of the contents of a system_log file:

```
20121231T205021|6|enable|Enabling the iridium modem
20121231T205024|6|network_quality|Modem quality = 5 on iridium
network.
20121231T205034|6|network_reg|Modem successfully registered
with iridium network. 13 tries remaining
20121231T205034|7|send_recv_cmd|Sending: AT &C1 &D2 &K0 &R1 &S1
E1 Q0 S0=0 S7=45 S10=100 V1 X4
20121231T205034|7|send_recv_cmd|Expecting: OK
```

SECTION 5: Modifying the Mission Plan

The APEX profiling float behaves in accordance with a factory programmed mission that can be modified by the operator either locally by connecting the float to a PC or remotely over the Iridium satellite network after the float is deployed.

NOTE *If not using a service provider, refer APPENDIX F: "Setting up Communications" to set up for remote communications with the float.*

A mission is modified by changing the settings of the mission.cfg file configuration parameters. These configuration parameters are downloaded by the float during the Mission Prelude phase and during each Surface phase. In addition, a number of user commands are available that can be used to control some of the float functions and access its files when it is connected to a PC.

NOTE *In addition to the mission plan, the sampling behavior of the float can be modified remotely by changing the settings of the configuration parameters in the sample.cfg file. Similar to that for a mission.cfg file, these settings are downloaded by the float during the Mission Prelude phase and during each Surface phase. The settings can also be edited locally by connecting a float to a PC. For instructions on how to edit these configuration parameter settings locally, refer to "Viewing and Editing the Configuration Parameters" on page 2-14.*

5.1 Modifying the mission.cfg File Configuration Parameter Settings Locally from a Connected PC

The configuration parameter settings in the mission.cfg file of the float can be modified directly by connecting a PC to the float and using a terminal program to enter commands.

CAUTION ***ParkDescentCount*** *and* ***DeepDescentCount*** *are dynamically updated during a mission. Therefore they should not be included in the modified mission.cfg file.*

NOTE *For ice avoidance some configuration parameters are hidden if they are not active.*

To modify the mission.cfg file configuration parameter settings locally from a connected PC:

1. Connect a PC to the float and start a terminal program as described in “Connecting a PC to the Float” on page 2-10.
2. Enter the configuration parameter to be changed and its value in accordance with the following format:

```
mission_set <Configuration Parameter Name> <Value>
```

NOTE *If <Value> is outside of the specified range for the configuration parameter, an error message will be displayed.*

3. Repeat Step 2 for all the configuration parameter settings that you want to change.
4. If desired, to display a specific configuration parameter setting, enter:

```
mission_get <Configuration Parameter Name>
```

5. If desired, to display all of the configuration parameter settings, enter:

```
mission_get
```

CAUTION *To save the changed configuration parameter settings, the **mission_save** command must be entered.*

6. To save the new setting or settings, enter:

```
mission_save
```

The float processes the mission.cfg file, checks for and displays errors if any and rejects the save operation if errors are found. An error will be found, for example, if **ParkPressure** is 1500 and **DeepDescentPressure** 1000. If no errors are found, the new settings are saved.

If desired, to modify all the configuration parameter settings to their default values, enter:

```
mission_default
```

For example, to change **ParkPressure** to 900 and then display and save the new setting, enter:

```
mission_set ParkPressure 900
mission_get ParkPressure
mission_save
```

For more information on these commands, refer to “User Commands” on page 5-5.

5.2 Modifying the mission.cfg File Configuration Parameter Settings Remotely

The configuration parameter settings in the mission.cfg file on the float can be modified remotely by modifying the mission.cfg file in the float's home directory on the server. After the float downloads the modified mission.cfg file, the file is recorded to the system_log.txt file, enabling verification by the operator when the log file is transmitted.

NOTE *The transmitted log file will include only the changed parameters.*

CAUTION ***ParkDescentCount*** *and* ***DeepDescentCount*** *are dynamically updated during a mission. Therefore they should not be included in the modified mission.cfg file.*

CAUTION *Some configuration parameters can conflict resulting in an error and the total rejection of the mission.cfg file. Be sure to verify as correct all changed configuration parameters when creating a new mission.cfg file. For example, setting* ***ParkPressure*** *to 1500 and* ***DeepDescentPressure*** *to 1000 will result in an error and the total rejection of any other changed configuration parameters. Should this situation occur, the existing mission.cfg file will be used and the error will be logged to the system_log.txt file.*

To modify the mission.cfg file configuration parameters remotely:

1. Using a text editor, such as Microsoft Notepad, open the mission.cfg file for editing in the float's home directory on the server.
2. Enter the configuration parameter to be changed and its value in accordance with the following format:

   ```
   <Configuration Parameter Name> <Value>
   ```

3. Repeat Step 2 for all the configuration parameter settings that you want to change.

   There must only be one configuration parameter and its setting for each line. In addition, it is only necessary for the mission.cfg file to include the changed configuration parameter settings. Unchanged settings need not be included in the file and can be deleted if present. Comments can also be included in the file by starting the line with a "#" character.

4. Save the modified mission.cfg file and place it into the float's home directory on the server.

   When the float surfaces, it will download the contents of the modified mission.cfg file. Once this occurs the file can be removed from the server so the float does not continue to download the same file each time it surfaces.

For example, to change **ParkPressure** to 900, enter:

```
ParkPressure 900
```

5.3 Modifying the sample.cfg File Configuration Parameter Settings Remotely

The configuration parameter settings in the sample.cfg file of the float can be modified remotely by modifying the sample.cfg file in the float's home directory on the server.

NOTE *The sample.cfg file must include all the required parameters, not just the changed ones. Leaving out an entry will result in it being deleted from the file.*

To modify the sample.cfg file remotely:

1. Using a text editor, such as Microsoft Notepad, open the sample.cfg file for editing in the float's home directory on the server.
2. Delete, change or add entries as described in "sample.cfg File" on page 4-2.
3. Save the modified sample.cfg file.

   When the float surfaces, it will download the contents of the modified sample.cfg file. Once this occurs the file can be removed from the server so the float does not continue to download the same file each time it surfaces.

For example, to sample the CTD every 100 dbar beginning at 2000 dbar and ending at 100 dbar, *add* the entry:

```
<ASCENT>
SAMPLE CTD 2000 100 100 DBAR
```

5.4 User Commands

The float includes a number of user commands that can be executed when a PC is connected to the float and is running a terminal program. These commands are divided into six main categories:

- File system
- Console
- Mission parameters
- Modem
- Mission
- System
- Device
- Test

5.4.1 File System Commands

The following File System commands are available:

fs_cat. Concatenates and displays the contents of a specific file.

Syntax: `fs_cat <filename>`

Example: `fs_cat system.cfg`

fs_cd. Changes the directory.

Syntax: `fs_cd <directory name>`

Example: `fs_cd download`

fs_cp. Copies the contents of a specified source file to a destination file.

Syntax: `fs_cp <source file> <destination file>`

Example: `fs_cp mission.cfg mission_temp.cfg`

fs_df. Displays file system disk space usage.

Syntax: `fs_df`

Example:

```
fs_df
512-blocks
Filesystem  Used  Available  Capacity  Mounted on
sdcard:0:   6976   15736064  15743040    sdcard:0:
```

fs_ls. Lists the files in a specified directory.

Syntax: `fs_ls`

Example:
```
fs_ls
-rw-rw-rw-     885 mar 22 3917 15:06 mission.cfg
```

fs_mv. Renames a file.

Syntax: `fs_mv` *<source file> <destination file>*

Example: `fs_mv mission.cfg mission.save`

fs_pwd. Displays the working directory.

Syntax: `fs_pwd`

Example:
```
fs_pwd
sdcard:0:\download
```

fs_rm. Removes a specified file.

Syntax: `fs_rm` *<filename>*

Example: `fs_rm mission.cfg`

fs_touch. Changes the timestamps of a specified file.

Syntax: `fs_touch` *<filename>*

Example: `fs_touch mission.cfg`

fs_wc. Displays the number of bytes, words and lines of a specified file.

Syntax: `fs_wc` *<filename>*

Example:
```
fs_wc mission.cfg
41          85        885 mission.cfg
```

5.4.2 Console Commands

The following Console command is available:

term_fc. Displays the console command input history.

Syntax: `term_fc`

Example:
```
term_fc
0     m_console
1     sys_date
2     m_state
3     mission_save
4     mission_get
```

5.4.3 Mission Parameters Commands

The following Mission Parameters commands are available:

mission_set. Sets a specified mission.cfg file configuration parameter to a specified setting. The setting is not saved until the **mission_save** command is entered.

Syntax: mission_set <*Configuration Parameter Name*> <*Value*>

Example:

```
mission_set ParkPressure 800
ParkPressure 800
```

NOTE *There are two copies of the mission.cfg file. One is the active copy, and the other is an editable, temporary copy. The* ***mission_get*** *command displays the configuration parameter settings of the temporary copy. The temporary copy becomes the active copy only when the* ***mission_save*** *command is entered, otherwise it is deleted on a system restart.*

mission_get. Displays the setting of a specified mission.cfg file configuration parameter of the temporary mission.cfg file. If no configuration parameter is specified, all of the settings in the temporary mission.cfg file are displayed.

Syntax: mission_get <*Configuration Parameter Name*>

Example:

```
mission_get ParkPressure
ParkPressure 800
```

mission_save. Processes the mission.cfg file and checks for errors. If no errors are found, the changed mission.cfg file configuration parameter settings are saved. If errors are found, the errors are displayed and the changed settings are not saved.

Syntax: mission_save

mission_print. Displays all of the active mission.cfg file configuration parameter settings.

Syntax: mission_print

mission_default. Sets the active mission.cfg file configuration parameters to their default settings.

Syntax: mission_default

Example:

```
mission_default
20200513T203546|5|T_CMD|mission_default
20200513T203547|5|mission_cfg|PreludeSelfTest off to on
20200513T203547|5|mission_cfg|PreludeTime 0 to 120
mission update to defaults PASSED
```

5.4.4 Modem Commands

The following Modem commands are available:

modem_transfer. Downloads mission.cfg and sample.cfg files if available from the server to the float or uploads all files from the upload directory of the float to the server. You can specify which, a download or an upload.

Syntax: `modem_transfer <up|down>`

log_up. Compresses and moves the current open log files to the float's upload directory for uploading to the server.

Syntax: `log_up`

modem_test. Tests the Iridium modem's primary and secondary dial up phone or RUDICS ID numbers.

Syntax: `modem_test <primary|secondary>`

Refer to APPENDIX H: "Diagnostic Messages Example" for an example output.

modem_sky_search. Attempts to register with the IRIDIUM network to determine if the float can complete telemetry.

Syntax: `modem_sky_search`

Example:

```
modem_sky_search
20180419T162635|5|T_CMD|modem_sky_search
20180419T162850|3|network_reg|Modem Registration Failed: 12 tries
20180419T162850|4|sky_search|No sky
Modem Sky Search: FAIL
```

modem_show. Displays the model number, the firmware version, the serial number, and other information pertaining to the Iridum modem that is installed in the float.

Syntax: `modem_show`

Example:

```
modem_show
20180509T191722|5|T_CMD|modem_show
20180509T191722|5|get_modem_info|Modem: Iridium
20180509T191722|5|get_modem_info|Model: 9523N
20180509T191723|5|get_modem_info|Call Processor Version: DB16009
20180509T191723|5|get_modem_info|Modem DSP Version: 1.7 svn: 2358
20180509T191723|5|get_modem_info|Audio DSP Version: 1.7 svn: 2459
20180509T191723|5|get_modem_info|DBB Version: 0x0001 (ASIC)
20180509T191723|5|get_modem_info|Main RFA Version: 0x0007 (SRFA2)
20180509T191723|5|get_modem_info|Aux RFA Version: 0x07ff (Not present)
20180509T191723|5|get_modem_info|NVM Version: KVS
20180509T191723|5|get_modem_info|BOOT Version:
BOOT0fa3/9523NRevE/02/RAW15
20180509T191723|5|get_modem_info|Serial No.: 300125060480510
20180509T191724|5|get_modem_info|SIM ICCID: 8988169312003720319
```

5.4.5 Mission Commands

The following Mission commands are available:

m_deploy. Runs the Mission Prelude followed by the start of the mission.

Syntax: `m_deploy`

m_bye. Exits Console mode, and after the PC is disconnected from the float, enables lowpower standby operation and sets the console verbosity back to 0.

Syntax: `m_bye`

m_state. Displays the current state of the float.

Syntax: `m_state`

m_console. Sets the console verbosity level to 5, disables lowpower standby for a default time of two hours and places the float into Console mode where it does not check for the pressure activation depth or run mission operations unless commanded to do so. The time may be specified to between 1 and 48 hours. Use the **m_bye** command to reset the verbosity to 0 and exit Console mode.

Syntax: `m_console <time>`

5.4.6 System Commands

The following System commands are available:

sys_grep. Searches for and displays lines containing a specified pattern of characters in a specified file.

Syntax: `sys_grep <pattern> <file>`

Example:
```
sys_grep Rate mission.cfg
AscentRate 0.08
```

sys_date. Displays the system date and time.

Syntax: `sys_date`

Example:
```
sys_date
Wed Mar 22 15:43:44 2017
```

sys_reboot. Executes a system reset that stops the running mission and returns the float to Idle mode.

Syntax: `sys_reboot`

sys_emerg_clr. Executes a system reset that stops the running mission and returns the float to Idle mode.

Syntax: sys_emerg_clr

NOTE *Before using the* ***sys_emerg_clr*** *command, contact Teledyne Webb Research customer service for recommendations, as sending this command will cause the float to reset and go into Idle mode where it will periodically check for its pressure activation depth.*

sys_ver. Displays the system firmware version.

Syntax: sys_ver

Example: sys_ver
03/21/17 22:37:56 APF11-v2.6.1

sys_self_test. Runs the self test without starting a mission. When the "-nosky" option is provided, the self test does not run gps or telemetry testing.

Syntax: sys_self_test <-nosky>

sys_test. Tests the sensor specified. Refer to APPENDIX A: "Optional Sensors." for information about testing each sensor.

Syntax: sys_test <*sensor*>

Example: sys_test OPT

sys_show. Displays all of the information available from the sensor.

Syntax: sys_show <*sensor*>

Example: sys_show OPT

sys_capture. Captures subsequent command line output into a specified file. Typing "." followed by pressing the Enter key at the end of the command line output ends the capture.

Syntax: sys_capture <*filename*>

sys_verbosity. Sets the verbosity of the console output over the serial interface to a specified level. The range is 0 (no output) to 6 (1 through 6 levels output) as follows:

0 – no outputs (default)
1 – critical errors
2 – major errors
3 – minor errors
4 – warnings
5 – notices
6 – informational

Syntax: `sys_verbosity`

Example: `sys_verbosity 5`

5.4.7 Device Commands

The following Device commands are available:

gps_update. Attempts to obtain a current GPS fix for time and location.

Syntax: `gps_update`

Example:
```
gps_update
20180419T162419|5|T_CMD|gps_update
Updating GPS time and location
```

pot_read. Displays the location of the buoyancy position in counts.

Syntax: `pot_read`

Example:
```
pot_read
20180419T162419|5|T_CMD|pot_read
Potentiometer Value: 2299
```

strobe. Interface to turn on (flash), off or test the strobe device.

Syntax: `strobe <on|off|test>`

Example:
```
> strobe on
20180419T163304|5|T_CMD|strobe on
20180419T163305|5|STROBE|on
> strobe off
20180419T163317|5|T_CMD|strobe off
20180419T163318|5|STROBE|off
> strobe test
20180419T163355|5|T_CMD|strobe test
20180419T163355|5|STROBE|test|start
20180419T163357|5|STROBE|on
20180419T163403|5|STROBE|off
20180419T163405|5|STROBE|on
20180419T163411|5|STROBE|off
20180419T163418|5|STROBE|on
20180419T163424|5|STROBE|off
20180419T163431|5|STROBE|on
20180419T163437|5|STROBE|off
```

```
20180419T163443|5|STROBE|on
20180419T163450|5|STROBE|off
20180419T163456|5|STROBE|on
20180419T163528|5|STROBE|off
20180419T163528|5|STROBE|test|PASSED
```

5.4.8 Test Commands

The following Device commands are available:

air_test. Tests the air engine system by running an inflation and deflation sequence while monitoring the system vitals.

Syntax: `air_test`

Example:

```
air_test
20180419T152032|5|T_CMD|air_test
Air Engine Test:
```

buoy_test. Tests the buoyancy engine system by extending and retracting the buoyancy position while monitoring system vitals.

Syntax: `buoy_test`

Example:

```
buoy_test
20180419T152215|5|T_CMD|buoy_test
Buoyancy Engine Test:
```

SECTION 6: CTDs

The Teledyne Webb Research APEX profiling float includes one of the following CTDs:

- SBE 41CP (Sea-Bird Scientific)
- SBE 41 (Sea-Bird Scientific)
- SBE 41N+pH (Sea-Bird Scientific)
- RBR*argo*^{3} C.T.D (RBR Ltd.)

For the CTD specifications, refer to the CTD manufacturer’s Web site.

NOTE *Information, including technical descriptions, provided in this section is obtained from the CTD manufacturer documentation and is subject to change without notice.*

The following pages contain information about each of these CTDs, including how to specify their hardware configuration, how to specify their sampling behavior along with examples, how to uncompress and read their data log files, how to use the **Show** command to display all of the information available from the specific CTD, and how to display the as-shipped factory configuration parameters for the CTD.

6.1 SBE 41CP CTD

The SBE 41CP CTD from Sea-Bird Scientific is a pumped type CTD that provides both interval sampling and continuous profiling of water pressure, temperature and salinity.

6.1.1 Specifying the SBE 41CP CTD Hardware Configuration

A line in the sensors.cfg file specifies the hardware configuration for the SBE 41CP CTD:

```
seabird_ctd_41cp COM2 B9600
```

For more information on the sensors.cfg file along with additional examples, refer to “sensors.cfg File” on page 4-16.

6.1.2 Specifying the SBE 41CP CTD Sampling Behavior

Interval sampling and continuous profiling, which can be performed separately or simultaneously, are controlled with entries in the sample.cfg file. In both cases the specified start and stop pressures can be in any order. For more information on using the sample.cfg file, including definitions and ranges for start, stop, interval, units, and count, along with additional examples, refer to “sample.cfg File” on page 4-2.

NOTE *For the best accuracy, it is recommended that interval sampling between 750 meters and the surface be performed simultaneously with continuous profiling, except when **tswait**=20 or greater.*

Interval sampling. The interval sampling behavior for the SBE 41CP CTD is specified using a single entry in the sample.cfg file for each of any number of pressure ranges in accordance with the following syntax:

<*phase*>
SAMPLE <*sensor type*> *[*<*start*> *[*<*stop*> *[*<*interval*> *[*<*units*> *[*<*count*>*]]]]]*

For example, to sample the SBE 41CP CTD PTS every 10 dbar beginning at 1000 dbar and ending at 200 dbar, enter:

```
<ASCENT>
SAMPLE PTS 1000 200 10 DBAR
```

For the Park phase, interval sampling can be performed by specifying a time interval or a sampling interval based on **ParkTimerInterval**.

For example, to sample the SBE 41CP CTD PTS during the Park phase at the rate specified by **ParkTimerInterval**, enter:

```
<PARK>
SAMPLE PTS
```

For example, to sample the SBE 41CP CTD PTS during the Park phase every 8 hours, enter:

```
<PARK>
SAMPLE PTS 2500 0 28000 SEC
```

Continuous profiling. The continuous profiling behavior for the SBE 41CP CTD is specified using one or more entries in the sample.cfg file for each of any number of pressure ranges in accordance with the following syntax:

<phase>
PROFILE *<sensor type> [<start> [<stop>]]*

Continuous profiling will automatically stop at the depth determined by **SurfacePressure** or pcutoff, whichever is deeper. The bin size and sample rate are fixed at 2 decibars and 1 sample per second, respectively.

For example, to sample the SBE 41CP CTD continuously beginning at 1000 dbar and ending at 4 dbar:

```
<ASCENT>
PROFILE CTD 1000 4
```

6.1.3 Reading the Science Log SBE 41CP CTD Data

SBE 41CP CTD data are provided in the science_log.bin file. For instructions on how to uncompress this file, refer to “Converting the science_log.bin and vitals_log.bin Files to CSV Files” on page 4-19.

The rows in the science_log.csv file that contain data that were recorded when interval sampling with an SBE 41CP CTD are the following:

CTD_P. All rows in the science_log.csv file with the first column label "CTD_P" contain SBE 41CP CTD pressure data. The columns are from left to right:

Time stamp:	UTC (YYYYMMDDTHHMMSS)
Pressure:	Pressure in decibars

CTD_PT. All rows in the science_log.csv file with the first column label "CTD_PT" contain SBE 41CP CTD pressure and temperature data. The columns are from left to right:

Time stamp:	UTC (YYYYMMDDTHHMMSS)
Pressure:	Pressure in decibars
Temperature:	Temperature in degrees Celsius

CTD_PTS. All rows in the science_log.csv file with the first column label "CTD_PTS" contain SBE 41CP CTD data that were recorded when interval sampling. The columns are from left to right:

Time stamp: UTC (YYYYMMDDTHHMMSS)

Pressure: Pressure in decibars

Temperature: Temperature in degrees Celsius

Salinity: Salinity in PSU

The rows in the science_log.csv file that contain data that were recorded when continuous profiling with an SBE 41CP CTD are the following:

CTD_bins. All rows in the science_log.csv file with the first column label "CTD_bins" contain SBE 41CP CTD bin averaging statistics. The columns are from left to right:

Time stamp: UTC (YYYYMMDDTHHMMSS)

Samples: Number of samples recorded during the mission

Bins: Number of bins recorded during the mission

Maxpress: Highest pressure in decibars recorded during the mission

CTD_CP. All rows in the science_log.csv file with the first column label "CTD_CP" contain bin averaged SBE 41CP CTD data.

NOTE *When continuous profiling with a float that uses an SBE 41CP CTD, the time stamps in the science_log.bin file are not the times at which the samples were actually taken. Rather the time stamps are the times at which the data were recorded to the science_log.bin file which occurs when continuous profiling stops.*

The columns are from left to right:

Time stamp: UTC (YYYYMMDDTHHMMSS)

Pressure: Bin averaged pressure in decibars

Temperature: Bin averaged temperature in degrees Celsius

Salinity: Bin averaged salinity in PSU

Samples: Number of samples averaged per bin

Four example rows of SBE 41CP CTD data extracted from a science_log.csv file are shown in Figure 6-1.

CTD_P	20170303T050158	3.85			
CTD_PTS	20170303T050208	4.6	11.919	34.7004	
CTD_bins	20170303T050218	12917	743	1483.59	

CTD_CP	20170303T050218	1475.99	1.3475	34.7083	22

Figure 6-1: Example Rows of SBE 41CP CTD Data in a science_log.csv File

For more information on reading the science_log.bin file along with additional examples, refer to “Reading the science_log.csv File” on page 4-19.

6.1.4 Show Command

The **Show** command displays all of the information available from the SBE 41CP CTD. To use the **Show** command, a PC must be connected to the float and running a terminal program.

To display the SBE 41CP CTD configuration information, enter:

```
sys_show CTD
```

The SBE 41CP CTD configuration will be displayed on the PC as in the following example:

```
Model:                   SBE 41CP
Version:                 7.2.5
Serial Number:           8377
Variant:                 CP UW
tswait:                  20
pcutoff:                 2.00
automatic bin averaging: no
nsamples:                158
nbins:                   49
top bin interval:        2
top bin size:            2
top bin max:             10
middle bin interval:     2
middle bin size:         2
middle bin max:          20
bottom bin interval:     2
bottom bin size:         2
include transition bin:  no
include samples per bin: yes
real-time output:        PTS

temperature:  12-Apr-16

  TA0:                   -8.482609e-04
  TA1:                    2.894297e-04
```

```
  TA2:                    -3.591653e-06
  TA3:                     1.443891e-07

conductivity:  12-Apr-16

  G:                      -9.955153e-01
  H:                       1.501197e-01
  I:                      -4.438523e-04
  J:                       5.533339e-05
  CPCOR:                  -9.570000e-08
  CTCOR:                   3.250000e-06
  WBOTC:                  -1.032698e-06
pressure S/N = 4924326, range = 2900 psia:  13-Apr-16

  PA0:                     2.820664e+00
  PA1:                     3.928967e-04
  PA2:                     7.270721e-14
  PTCA0:                   1.077206e+05
  PTCA1:                   9.338340e+01
  PTCA2:                   1.166206e+01
  PTCB0:                   1.052019e+02
  PTCB1:                  -2.697820e-03
  PTCB2:                   0.000000e+00
  PTHA0:                   3.179357e+02
  PTHA1:                  -9.059575e-05
  PTHA2:                   2.859202e-12
  POFFSET:                 0.000000e+00
```

6.1.5 SBE 41CP Commands

The following SBE 41CP commands are available:

ctd_config. Runs the configuration sequence for the SBE 41CP CTD.

Syntax: `ctd_config`

ctd_p. Obtains sample pressure data from the SBE 41CP CTD.

Syntax: `ctd_p`

ctd_pt. Obtains sample pressure and temperature data from the SBE 41CP CTD.

Syntax: `ctd_pt`

ctd_pts. Obtains sample pressure, temperature and salinity data from the SBE 41CP CTD.

Syntax: `ctd_pts`

ctd_show. Displays all of the sensor information available from the SBE 41CP CTD.

Syntax: `ctd_show`

ctd_test. Runs through configuration, displays configuration and obtains sample data from the SBE 41CP CTD.

Syntax: `ctd_test`

6.1.6 SBE 41CP CTD Factory Configuration

The factory configuration of the installed SBE 41CP CTD is the following:

```
autobinavg = 0
top_bin_interval = 2
top_bin_size = 2
top_bin_max = 10
middle_bin_interval = 2
middle_bin_size = 2
middle_bin_max = 20
bottom_bin_interval = 2
bottom_bin_size = 2
includetransitionbin = 0
includenbin = 1
outputpts = 1
tswait = 20
pcutoff = 2
```

6.2 SBE 41 CTD

The SBE 41 CTD from Sea-Bird Scientific is a pumped type CTD that provides interval sampling of water pressure, temperature and salinity.

6.2.1 Specifying the SBE 41 CTD Hardware Configuration

A line in the sensors.cfg file specifies the hardware configuration for the SBE 41 CTD:

```
seabird_ctd_41 COM2 B9600
```

For more information on the sensors.cfg file along with additional examples, refer to “sensors.cfg File” on page 4-16.

6.2.2 Specifying the SBE 41 CTD Sampling Behavior

Sampling is controlled with entries in the sample.cfg file. The specified start and stop pressures can be in any order. For more information on using the sample.cfg file, including definitions and ranges for start, stop, interval, units, and count, along with additional examples, refer to “sample.cfg File” on page 4-2.

The sampling behavior for the SBE 41 CTD is specified using a single entry in the sample.cfg file for each of any number of pressure ranges in accordance with the following syntax:

<phase>
SAMPLE *<sensor type> [<start> [<stop> [<interval> [<units> [<count>]]]]]*

For example, to sample the SBE 41 CTD every 10 dbar beginning at 1000 dbar and ending at 200 dbar, enter:

```
<ASCENT>
SAMPLE CTD 1000 200 10 DBAR
```

For the Park phase, interval sampling can be performed by specifying a time interval or a sampling interval based on **ParkTimerInterval**.

For example, to sample the SBE 41 CTD PTS during the Park phase at the rate specified by **ParkTimerInterval**, enter:

```
<PARK>
SAMPLE PTS
```

For example, to sample the SBE 41 CTD PTS during the Park phase every 8 hours, enter:

```
<PARK>
SAMPLE PTS 2500 0 28000 SEC
```

6.2.3 Reading the Science Log SBE 41 CTD Data

SBE 41 CTD data are provided in the science_log.bin file. For instructions on how to uncompress this file, refer to “Converting the science_log.bin and vitals_log.bin Files to CSV Files” on page 4-19.

With the exception of CTD_bins and CTD_CP, which are not included, the rows in the science_log.csv file for an SBE 41 CTD are the same as that provided for an SBE 41CP CTD as described in “Reading the Science Log SBE 41CP CTD ” on page 6-3.

Two example rows of SBE 41 CTD data extracted from a science_log.csv file are shown in Figure 6-2.

CTD_P	20170303T050158	3.85			
CTD_PTS	20170303T050208	4.6	11.919	34.7004	

Figure 6-2: Example Rows of SBE 41 CTD Data in a science_log.csv File

For more information on reading the science_log.bin file along with additional examples, refer to “Reading the science_log.csv File” on page 4-19.

6.2.4 Show Command

The **Show** command displays all of the information available from the SBE 41 CTD. To use the **Show** command, a PC must be connected to the float and running a terminal program.

To display the SBE 41 CTD configuration information, enter:

```
sys_show CTD
```

The SBE 41 CTD configuration will be displayed on the PC as in the following example:

```
Model:                   SBE 41
Version:                 7.2.5
Serial Number:           9565
Variant:                 Standard
tswait:                  2
temperature:  21-Mar-17
  TA0:                   -8.177796e-04
  TA1:                   2.853510e-04
  TA2:                   -3.361656e-06
  TA3:                   1.382331e-07
conductivity:  21-Mar-17
  G:                     -1.004127e+00
```

```
  H:                     1.414276e-01
  I:                     -3.575581e-04
  J:                     4.619621e-05
  CPCOR:                 -9.570000e-08
  CTCOR:                 3.250000e-06
  WBOTC:                 9.360629e-08
pressure S/N = 10449802, range = 2900 psia:  14-Mar-17
  PA0:                   7.541767e-01
  PA1:                   3.932653e-04
  PA2:                   -3.032656e-13
  PTCA0:                 1.680632e+03
  PTCA1:                 2.568692e+01
  PTCA2:                 2.626050e+00
  PTCB0:                 2.448937e+01
  PTCB1:                 -2.125000e-03
  PTCB2:                 0.000000e+00
  PTHA0:                 3.275166e+02
  PTHA1:                 -6.205709e-05
  PTHA2:                 -1.461976e-12
  POFFSET:               0.000000e+00
```

6.2.5 SBE 41 CTD Commands

The following SBE 41 CTD commands are available:

ctd_config. Runs the configuration sequence for the SBE 41 CTD.

Syntax: `ctd_config`

ctd_p. Obtains sample pressure data from the SBE 41 CTD.

Syntax: `ctd_p`

ctd_pt. Obtains sample pressure and temperature data from the SBE 41 CTD.

Syntax: `ctd_pt`

ctd_pts. Obtains sample pressure, temperature and salinity data from the SBE 41 CTD.

Syntax: `ctd_pts`

ctd_show. Displays all of the sensor information available from the SBE 41 CTD.

Syntax: `ctd_show`

ctd_test. Runs through configuration, displays configuration and obtains sample data from the SBE 41 CTD.

Syntax: `ctd_test`

6.2.6 SBE 41 CTD Factory Configuration

The factory configuration of the installed SBE 41 CTD is the following:

```
tswait = 2
```

6.3 SBE 41N+pH CTD

The SBE 41N+pH CTD from Sea-Bird Scientific is a pumped type CTD that provides both interval sampling and continuous profiling of water pressure, temperature, salinity, and pH. A Sea-Bird Scientific SeaFET Ocean pH sensor provides the pH measurements. This sensor is an ion selective field effect transistor (ISFET) type device for accurate long term pH measurements in salt water. Refer also to Sea-Bird Electronics document “Quick Start Guide–Deep-Sea ISFET pH sensor integrated with a 41N profiling float CTD.”

6.3.1 Specifying the SBE 41N+pH CTD Hardware Configuration

A line in the sensors.cfg file specifies the SBE 41N+pH CTD hardware configuration:

```
seabird_ctd_41n COM2 B9600
```

For more information on the sensors.cfg file along with additional examples, refer to “sensors.cfg File” on page 4-16.

6.3.2 Specifying the SBE 41N+pH CTD Sampling Behavior

Interval sampling and continuous profiling are controlled with entries in the sample.cfg file. In both cases the specified start and stop pressures can be in any order. For more information on using the sample.cfg file, including definitions and ranges for start, stop, interval, units, and count, along with additional examples, refer to “sample.cfg File” on page 4-2.

NOTE *For the best accuracy, it is recommended that interval sampling between 750 meters and the surface be performed simultaneously with continuous profiling, except when tswait=20 or greater.*

Interval sampling. The interval sampling behavior for the SBE 41N+pH is specified using one or more entries in the sample.cfg file for each of any number of pressure ranges in accordance with the following syntax:

<*phase*>
SAMPLE <*sensor type*> *[*<*start*> *[*<*stop*> *[*<*interval*> *[*<*units*> *[*<*count*>*]]]]]*

For example, to sample *both* the CTD and the pH sensor every 10 dbar beginning at 1000 dbar and ending at 200 dbar, enter:

```
<ASCENT>
SAMPLE PH 1000 200 10 DBAR
```

To sample *only* the CTD every 10 dbar beginning at 1000 dbar and ending at 200 dbar, enter:

```
<ASCENT>
SAMPLE CTD 1000 200 10 DBAR
```

To sample *both* the CTD and the pH sensor every 10 dbar beginning at 600 dbar and ending at 4 dbar and *only* the CTD every 20 dbar beginning at 2000 dbar and ending at 1000 dbar, enter:

```
<ASCENT>
SAMPLE CTD 2000 1000 20 DBAR
SAMPLE PH 600 4 10 DBAR
```

For the Park phase, interval sampling can be performed by specifying a time interval or a sampling interval based on **ParkTimerInterval** for either the CTD or both the CTD and the pH sensor.

For example, to sample the SBE 41N+pH CTD PTS during the Park phase at the rate specified by **ParkTimerInterval**, enter:

```
<PARK>
SAMPLE PTS
```

To sample the SBE 41N+pH CTD PTS during the Park phase every 8 hours, enter:

```
<PARK>
SAMPLE PTS 2500 0 28000 SEC
```

To sample *both* the CTD and the pH sensor during the Park phase, enter:

```
<PARK>
SAMPLE PH 2500 0 28000 SEC
```

NOTE *Interval sampling of pH cannot be performed during continuous CTD or pH profiling. Entries of this type will be ignored. Furthermore, continuous pH profiling can only be performed when performing continuous CTD profiling.*

Continuous CTD profiling. The continuous CTD profiling behavior for the SBE 41N+pH CTD is specified using one or more entries in the sample.cfg file for each of any number of pressure ranges in accordance with the following syntax:

<phase>
PROFILE CTD *[<start> [<stop>]]*

Continuous profiling will automatically stop at the depth determined by **SurfacePressure** or pcutoff. The bin size and sample rate are fixed at 2 decibars and 1 sample per second, respectively.

For example, to sample the SBE 41N+pH continuously beginning at 1000 dbar and ending at 4 dbar:

```
<ASCENT>
PROFILE CTD 1000 4
```

Continuous pH profiling. The continuous pH profiling behavior for the SBE 41N+pH CTD is specified using one or more entries in the sample.cfg file for each of any number of pressure ranges in accordance with the following syntax:

<phase>
PROFILE PH *[<start> [<stop> [<interval>]]]*

If no entry is provided for <interval>, a default of 1 second is used.

For example, to sample the CTD continuously beginning at 1000 dbar and ending at 4 dbar and the pH sensor beginning at 1000 dbar and ending at 4 dbar with a sampling interval of 2 seconds, enter:

```
<ASCENT>
PROFILE CTD 1000 4
PROFILE PH 1000 4 2
```

Both entries following the phase entry are required. In this example continuous profiling for the CTD and the pH sensor is performed between the same start and stop pressures, and the fixed bin size and sample rate of 2 decibars and 1 sample per second, respectively, are used for the CTD.

To sample the CTD continuously beginning at 2000 dbar and ending at 4 dbar and the pH sensor beginning at 500 dbar and ending at 4 dbar with a sampling interval of 2 seconds, enter:

```
<ASCENT>
PROFILE CTD 2000 4
PROFILE PH 500 4 2
```

Again, both entries following the phase entry are required. In this example continuous profiling for the CTD and the pH sensor are performed beginning at different start pressures and ending at the same stop pressure.

NOTE *By default, the SBE 41N+pH CTD is internally configured to start profiling at 1000 decibars, to stop profiling at 0 decibars and to sample once per second. However, continuous pH profiling can only occur during continuous CTD profiling. When continuous CTD profiling is specified without specifying pH profiling, pH samples will be included when the default pH profiling range coincides with the specified continuous CTD profiling range.*

In the following example the CTD is sampled continuously beginning at 1500 dbar and ending at 4 dbar along with the pH sensor beginning by default at 1000 dbar and also ending at 4 dbar, and with a sampling interval of 1 second.

```
<ASCENT>
PROFILE CTD 1500 4
```

In the following example the CTD is sampled continuously beginning at 2000 dbar and ending at 1000 dbar along with the pH sensor beginning at 1500 dbar and also ending at 1000 dbar with a sampling interval of 1 second. Continuous sampling of the pH sensor is ignored after 1000 dbar even though 500 dbar is specified, as continuous CTD profiling is not being performed in this pressure ranged.

```
<ASCENT>
PROFILE CTD 2000 1000
PROFILE PH 1500 500 1
```

NOTE *Only one entry of sensor type PH is allowed. If more than one entry of this sensor type is included, the sample.cfg file will be rejected, and an error message will be logged to the system_log.txt file.*

To override the default configuration of the pH sensor and therefore perform no pH sampling, yet sample the CTD as in the above example, enter:

```
<ASCENT>
PROFILE CTD 2000 1000
PROFILE PH 0 0 1
```

6.3.3 Reading the Science Log SBE 41N+pH CTD Data

SBE 41N+pH CTD data are provided in the science_log.bin file. For instructions on how to uncompress this file, refer to “Converting the science_log.bin and vitals_log.bin Files to CSV Files” on page 4-19.

The rows in the science_log.csv file for an SBE 41N+pH CTD are the same as that provided for an SBE 41CP CTD as described in “Reading the Science Log SBE 41CP CTD ” on page 6-3 plus the additional rows listed below.

CTD_PTSH. All rows in the science_log.csv file with the first column label "CTD_PTSH" contain SBE 41N+pH CTD pressure, temperature, salinity, and pH data that were recorded when interval sampling. The columns are from left to right:

Time stamp:	UTC (YYYYMMDDTHHMMSS)
Pressure:	Pressure in decibars
Temperature:	Temperature in degrees Celsius
Salinity:	Salinity in PSU
pH:	Reference voltage

CTD_CP_H. All rows in the science_log.csv file with the first column label "CTD_CP_H" contain bin averaged SBE 41N+pH CTD data that were recorded when continuous profiling.

NOTE *When continuous profiling with a float that uses an SBE 41N+pH CTD, the time stamps in the science_log.bin file are not the times at which the samples were actually taken. Rather the time stamps are the times at which the data were recorded to the science_log.bin file which occurs when continuous profiling stops.*

The columns are from left to right:

Time stamp:	UTC (YYYYMMDDTHHMMSS)
Pressure:	Bin averaged pressure in decibars
Temperature:	Bin averaged temperature in degrees Celsius
Salinity:	Bin averaged salinity in PSU
PTS samples:	Number of PTS samples averaged in a bin
pH:	Reference voltage
pH samples:	Number of pH samples averaged in a bin

Two example rows of SBE 41N+pH CTD data extracted from a science_log.csv file are shown in Figure 6-3.

CTD_PTSH	20170310T130912	91.6	21.9224	33.504	-0.960061		
CTD_CP_H	20170310T131412	4.17	21.9914	35.2916	23	-0.960538	23

Figure 6-3: Example Rows of SBE 41N+pH CTD Data in a science_log.csv File

For more information on reading the science_log.bin file along with additional examples, refer to "Reading the science_log.csv File" on page 4-19.

6.3.4 Show Command

The **Show** command displays all of the information available from the SBE 41N+pH CTD. To use the **Show** command, a PC must be connected to the float and running a terminal program.

To display the SBE 41N+pH CTD configuration information, enter:

```
sys_show CTD
```

The SBE 41N+pH CTD configuration will be displayed on the PC as in the following example:

```
Model:                   SBE 41N
Version:                 5.3.0
Serial Number:           9425
Variant:                 41cp
pcutoff:                 2.00
automatic bin averaging: no
nsamples:                283
nbins:                   0
top bin interval:        2
top bin size:            2
top bin max:             10
middle bin interval:     2
middle bin size:         2
middle bin max:          20
bottom bin interval:     2
bottom bin size:         2
include transition bin:  no
include samples per bin: yes
pump sample wait time:   20 sec
real-time output:        PTS
longpt:                  1
enableph:                1
measintervalph:          1
pressureonph:            1000
pressureoffph:           0
temperature_1:   07-Feb-17
  TA0:                   -8.423786e-04
  TA1:                   2.943042e-04
  TA2:                   -3.928960e-06
  TA3:                   1.509667e-07
temperature_2:   07-Feb-17
  TA0:                   -8.163393e-04
  TA1:                   2.916159e-04
  TA2:                   -3.680900e-06
  TA3:                   1.489342e-07
conductivity:  07-Feb-17
  G:                     -9.978469e-01
  H:                     1.514799e-01
  I:                     -3.834221e-04
```

```
  J:                       5.074354e-05
  CPCOR:                   -9.570000e-08
  CTCOR:                   3.250000e-06
  WBOTC:                   -7.126914e-07
pressure S/N = 10474358, range = 2900 psia:  02-Feb-17
  PA0:                     7.433826e-02
  PA1:                     3.887642e-04
  PA2:                     -2.891686e-13
  PTCA0:                   -4.937697e+03
  PTCA1:                   1.043480e+01
  PTCA2:                   -7.120038e-02
  PTCB0:                   2.535888e+01
  PTCB1:                   -2.500000e-05
  PTCB2:                   0.000000e+00
  PTHA0:                   -5.976322e+01
  PTHA1:                   5.322828e-02
  PTHA2:                   -3.091781e-07
  POFFSET:                 0.000000e+00
```

6.3.5 SBE 41N+pH CTD Commands

The following SBE 41N+pH CTD commands are available:

ctd_config. Runs the configuration sequence for the SBE 41N+pH CTD.

Syntax: `ctd_config`

ctd_p. Obtains sample pressure data from the SBE 41N+pH CTD.

Syntax: `ctd_p`

ctd_pt. Obtains sample pressure and temperature data from the SBE 41N+pH CTD.

Syntax: `ctd_pt`

ctd_pts. Obtains sample pressure, temperature and salinity data from the SBE 41N+pH CTD.

Syntax: `ctd_pts`

ctd_ptsh. Obtains sample pressure, temperature, salinity, and pH data from the SBE 41N+pH CTD.

Syntax: `ctd_ptsh`

ctd_show. Displays all of the sensor information available from the SBE 41N+pH CTD.

Syntax: `ctd_show`

ctd_test. Runs through configuration, displays configuration and obtains sample data from the SBE 41N+pH CTD.

Syntax: `ctd_test`

6.3.6 SBE 41N+pH CTD Factory Configuration

The factory configuration of the installed SBE 41N+pH CTD is the following:

```
outputpts=1
includetransitionbin=0
includenbin=1
autobinavg=0
tswait=20
top_bin_interval=2
top_bin_size=2
top_bin_max=10
middle_bin_interval=2
middle_bin_size=2
middle_bin_max=20
bottom_bin_interval=2
bottom_bin_size=2
pcutoff=2.000000
setlongpt=1
setenableph=1
setmeasintervalph=1
setpressureonph=1000
setpressureoffph=0
```

6.4 RBR*argo*^{3} C.T.D

The RBR*argo*^{3} C.T.D from RBR Ltd. is a non-pumped type CTD that provides both interval sampling and continuous profiling of water pressure, temperature, salinity, and conductivity.

6.4.1 Specifying the RBR*argo*^{3} C.T.D Hardware Configuration

A line in the sensors.cfg file specifies the hardware configuration for the RBR*argo*^{3} C.T.D:

```
sensor rbr_logger COM2 B19200
```

For more information on the sensors.cfg file along with additional examples, refer to “sensors.cfg File” on page 4-16.

6.4.2 Specifying the RBR*argo*^{3} C.T.D Sampling Behavior

Interval sampling and continuous profiling are controlled with entries in the sample.cfg file. In both cases the specified start and stop pressures can be in any order. For more information on using the sample.cfg file, including definitions and ranges for start, stop, interval, units, and count, along with additional examples, refer to “sample.cfg File” on page 4-2.

Interval sampling. The interval sampling behavior for the RBR*argo*^{3} C.T.D is specified using a single entry in the sample.cfg file for each of any number of pressure ranges in accordance with the following syntax:

<*phase*>
SAMPLE *<sensor type> [<start> [<stop> [<interval> [<units> [<count>]]]]]*

For example, to sample the RBR*argo*^{3} C.T.D PTS every 10 dbar beginning at 2000 dbar and ending at 0 dbar, enter:

```
<ASCENT>
SAMPLE PTS 2000 0 10 DBAR 1
```

For the Park phase, interval sampling can be performed by specifying a time interval or a sampling interval based on **ParkTimerInterval**.

For example, to sample the RBR*argo*^{3} C.T.D PTS during the Park phase at the rate specified by **ParkTimerInterval**, enter:

```
<PARK>
SAMPLE PTS
```

For example, to sample the RBR*argo*^{3} C.T.D CTD PTS during the Park phase every 8 hours, enter:

```
<PARK>
SAMPLE PTS 2500 0 28000 SEC
```

Continuous profiling. The continuous profiling behavior for the RBR*argo*^{3} C.T.D is specified using one or more entries in the sample.cfg file for each of any number of pressure ranges in accordance with the following syntax:

<phase>
PROFILE *<sensor type> [<start> [<stop> [<bin_size> [<rate>]]]]*

For example, to sample the RBR*argo*^{3} C.T.D continuously beginning at 2000 dbar and ending at 0 dbar with a bin size of 2 dbars and a sample rate of 1 sample/second, enter:

```
<ASCENT>
PROFILE LGR 2000 0 2 1 HZ
```

The RBR*argo*^{3} C.T.D can provide multiple channels of data. Other sensor type options that may be specified include P, PT, PTS, PTSC and PTSCI for pressure, temperature, salinity, conductivity and internal conductivity temperature channels.

6.4.3 Reading the Science Log RBR*argo*^{3} C.T.D Data

RBR*argo*^{3} C.T.D data are provided in the science_log.bin file. For instructions on how to uncompress this file, refer to “Converting the science_log.bin and vitals_log.bin Files to CSV Files” on page 4-19.

The rows in the science_log.csv file for an RBR*argo*^{3} C.T.D are listed below.

LGR_PTS. All rows in the science_log.csv file with the first column label "LGR_PTS" contain CTD data that were recorded when interval sampling. The columns are from left to right:

Time stamp:	UTC (YYYYMMDDTHHMMSS)
Pressure:	Pressure in decibars
Temperature:	Temperature in degrees Celsius
Salinity:	Salinity in PSU

LGR_PTSCI. All rows in the science_log.csv file with the first column label "LGR_PTSCI" contain CTD data that were recorded when interval sampling. The columns are from left to right:

Time stamp:	UTC (YYYYMMDDTHHMMSS)
Pressure:	Pressure in decibars
Temperature:	Temperature in degrees Celsius
Salinity:	Salinity in PSU
Conductivity:	Conductivity in mS/cm
Internal conductivity temperature:	Inside temperature in degrees Celsius

LGR_CP_PTSCI. All rows in the science_log.csv file with the first column label "LGR_CP_PTSCI" contain bin averaged RBR*argo*^{3} C.T.D data that were recorded when continuous profiling. The columns are from left to right:

Time stamp:	UTC (YYYYMMDDTHHMMSS)
Pressure:	Bin averaged pressure in decibars
Temperature:	Bin averaged temperature in degrees Celsius
Salinity:	Bin averaged salinity in PSU
Conductivity:	Conductivity in mS/cm
Internal conductivity temperature:	Inside temperature in degrees Celsius
Samples:	Number of samples averaged per bin

NOTE *When continuous profiling with a float that uses an RBRargo*^{3} *C.T.D, the time stamps are provided by the CTD, and they are the times at which the samples were taken at depth.*

Two example rows of RBR*argo*^{3} C.T.D data extracted from a science_log.csv file are shown in Figure 6-4.

LGR_PTS	20191130T132849	1004.2923	7.97122	35.15642			
LGR_PTSCI	20191130T132848	1004.25848	7.97122	35.15642	36.77597	7.93509	
LGR_CP_PTSCI	20191202T063830	198.22058	19.35117	36.66944	49.35155	19.31342	5

Figure 6-4: Example Rows of RBR*argo*^{3} C.T.D Data in a science_log.csv File

For more information on reading the science_log.bin file along with additional examples, refer to “Reading the science_log.csv File” on page 4-19.

6.4.4 Show Command

The **Show** command displays all of the information available from the RBR*argo*^{3} C.T.D. To use the **Show** command, a PC must be connected to the float and running a terminal program.

To display the RBR*argo*^{3} C.T.D configuration information, enter:

```
sys_show rbr_logger
```

The RBR*argo*[3] C.T.D configuration will be displayed on the PC as in the following example:

```
Model:                  RBRargo3
Fw Type:                104
Fw Version:             1.095
Serial No.:             203413
Power Info:             source = ext, int =  0.01, ext = 15.62,
reg = n/a
Baud Rate:              19200
Clock:                  datetime = 20200417170621, offsetfromutc =
+0.00
Power Used:             313.1e+000
Part Number:            0006298revA
Prof Periods:           500|250|125
Available Channels:   8
Enabled Channels:     6
Enabled Channels List:
channel 1: conductivity(mS/cm)
channel 2: temperature(C)
channel 3: pressure(dbar)
channel 4: salinity(PSU)
channel 5: measurement_count(counts)
channel 6: temperature(C)
Regimes: direction = ascending, count = 1, reference =
seapressure
Regime 1: boundary = 1000, binsize = 2.0, samplingperiod = 1000
```

6.4.5 RBR*argo*[3] C.T.D Commands

The following RBR*argo*[3] C.T.D commands are available:

sys_config. Runs the configuration sequence for the RBR*argo*[3] C.T.D.

Syntax: `sys_config rbr_logger`

sys_sample. Obtains sample pressure, temperature, salinity, conductivity, and internal temperature data from the RBR*argo*[3] C.T.D.

Syntax: `sys_sample rbr_logger`

sys_show. Displays all of the CTD Sensor information available from the RBR*argo*[3] C.T.D.

Syntax: `sys_show rbr_logger`

sys_test. Runs through configuration, displays configuration and obtains sample data from the RBR*argo*[3] C.T.D.

Syntax: `sys_test rbr_logger`

6.4.6 SBE RBR*argo*^{3} C.T.D Factory Configuration

The factory configuration of the installed RBR*argo*^{3} C.T.D is the following:

```
confirmation state = on
streamserial state = off
serial baudrate = 19200
clock datetime = 20170426153828
offsetfromut c = +0.00

channel 1 status = on
channel 1 conductivity_00 on
channel 2 status = on
channel 2 temperature_00 on
channel 3 status = off
channel 3 pressure_00 off
channel 4 status = on
channel 4 seapressure_00 on
channel 5 status = off
channel 5 depth_00 off
channel 6 status = on
channel 6 salinity_00 on
channel 7 status = on
channel 7 count_00 on
channel 8 status = on
channel 8 conductivitycelltemperature_00 on

6/8 channels enabled
outputformat type = caltext03
memformat newtype = calbin00
sampling mode = regimes
sampling period = 1000
regimes direction = ascending
regimes reference = seapressure
permit command = settings
settings atmosphere = 10.1325
```

SECTION 7: General Maintenance

After recovering the Teledyne Webb Research APEX profiling float, it should be cleaned and inspected. The battery usage should also be checked and the batteries replaced if the remaining capacity is not adequate for the next deployment. In addition, the CTD and any other science sensors should be checked and recalibrated if necessary.

7.1 Cleaning and Inspecting the Float after Recovery

After recovering a float, wash it with clean, fresh water and inspect the housing and the CTD for damage or corrosion. Also check that the zinc anode on the top end cap is secure and is not excessively corroded. The zinc anode is shown in Figure 1-2 on page 1-3. It should be replaced before redeploying the float. For instructions on how to replace the zinc anode, contact Teledyne Webb Research customer service.

7.2 Replacing the Float Batteries

For battery replacement return the float to Teledyne Webb Research.

WARNING *Do not attempt to open the APEX profiling float housing before returning the float to Teledyne Webb Research. If the float was recovered from the ocean, it may contain water. This situation presents a safety hazard due to the possible chemical reaction of the batteries with the water. Before returning a float to Teledyne Webb Research, contact Teledyne Webb Research customer service for instructions on how to properly prepare the float for shipment.*

7.3 Battery Usage

Battery usage is determined by the coulomb counter which indicates the current battery usage in mA-hours. This information is provided in the vitals_log.bin file as described in "vitals_log.bin File" on page 4-18. This file must be uncompressed and then converted to a CSV file before it can be opened in a spreadsheet program and read as described in "Converting the science_log.bin and vitals_log.bin Files to CSV File" on page 4-19.

NOTE *For an APEX profiling float with a CTD only and lithium batteries, the battery capacity is 108,000 mA-hours. For the same float with alkaline batteries, the battery capacity is 48,000 mA-hours. Floats with additional sensors may have different battery configurations and capacities.*

7.4 Storing the Float

Floats should be stored in a controlled environment of between 10°C to 25°C. For a float with a pumped type CTD, it should be stored in its original crate with the CTD inside the plastic bag and the red cap and the two orange plugs installed.

NOTE *If a float has been stored at a temperature of -2°C or less, allow it to warm up indoors before deployment.*

APPENDIX A: Optional Sensors

The Teledyne Webb Research APEX profiling float optionally includes one or more science sensors in addition to the CTD. Shown in Figure A-1 is an APEX-AMS Advanced Multisensor profiling float with several optional sensors installed. For the specifications for these sensors, refer to the sensor manufacturer's Web site.

NOTE *Information, including technical descriptions, provided in this section is obtained from the sensor manufacturer documentation and is subject to change without notice.*

The following pages contain information about each of the available optional sensors, including how to specify their hardware configuration, how to specify their sampling behavior along with examples, how to uncompress and read their data log files, how to use the **Show** command to display all of the information available from the specific sensor, and how to display the as-shipped factory configuration parameters for the sensor.

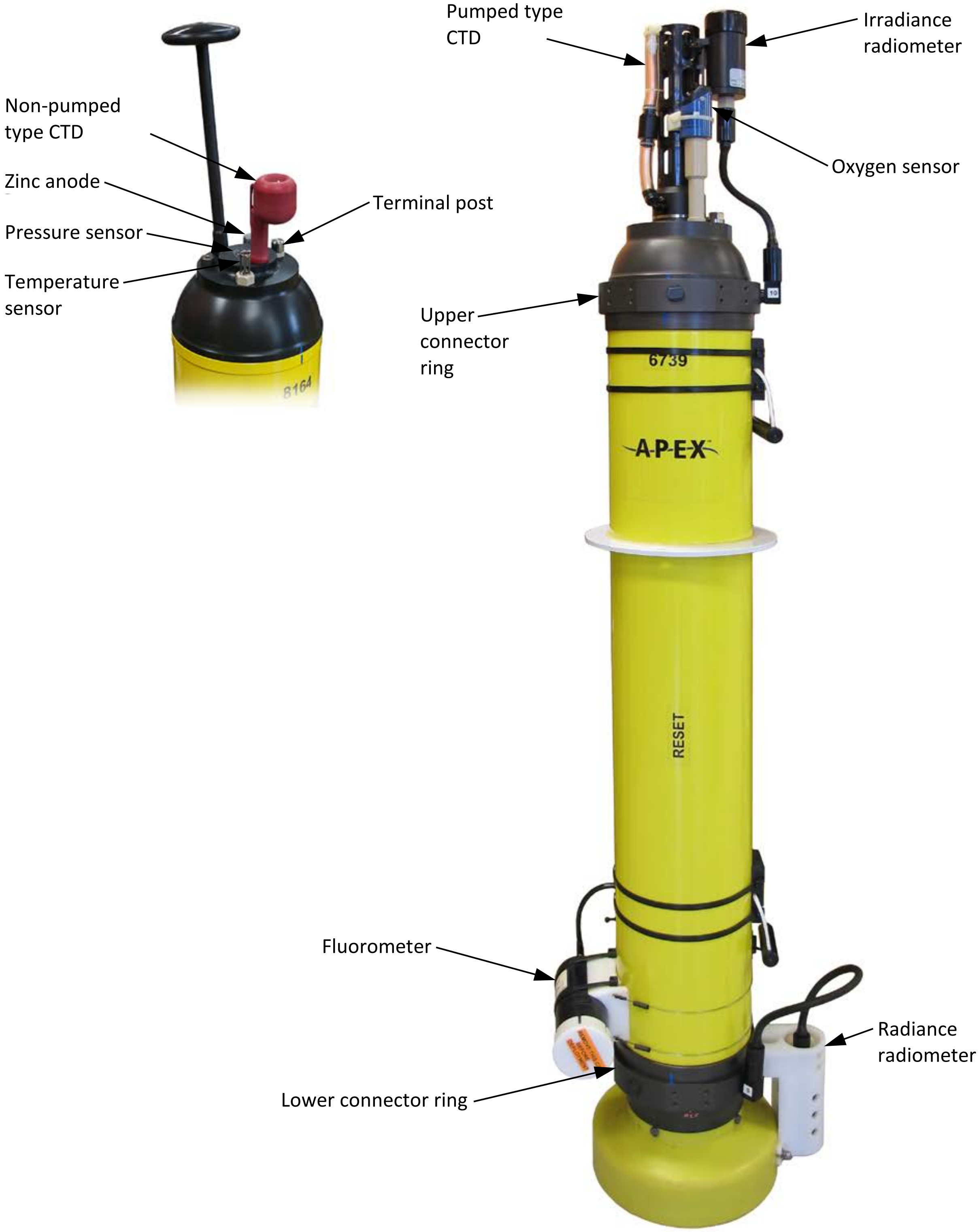


Figure A-1: The APEX Advanced Multisensor (AMS) Profiling Float with Pumped Type CTD and Non-pumped Type CTD

A.1 Oxygen Sensor (Aanderaa)

The Teledyne Webb Research APEX Deep profiling float optionally includes the Aanderaa Oxygen Optode 4330 Oxygen Sensor for periodic sampling of dissolved oxygen using an optical optode that does not consume oxygen. The measurement principle of the sensor is based on fluorescence quenching. The optode's foil is excited by modulated blue light, and the sensor measures the phase of a returned red light. By linearizing and temperature compensating with an incorporated temperature sensor located next to the sensing foil, the absolute O_2 concentration can be determined.

A.1.1 Specifying the Oxygen Sensor Hardware Configuration

A line in the sensors.cfg file defines the oxygen sensor hardware configuration as in the following example:

```
Optode COM4 B9600
```

For more information on the sensors.cfg file along with additional examples, refer to "sensors.cfg File" on page 4-16.

A.1.2 Specifying the Oxygen Sensor Sampling Behavior

The interval sampling behavior for the oxygen sensor is specified using a single entry in the sample.cfg file for each of any number of pressure ranges in accordance with the following syntax:

SAMPLE OPT *[<start> [<stop> [<interval> [<<units> [<count>]]]]]*

For example, to sample the oxygen sensor every 10 dbar beginning at 5000 dbar and ending at 200 dbar, enter:

```
<ASCENT>
SAMPLE OPT 5000 200 10 DBAR
```

The start and stop pressures can be in any order. For more information on using the sample.cfg file to program the behavior of the float along with additional examples, refer to "sample.cfg File" on page 4-2.

A.1.3 Specifying the Oxygen Sensor In-Air Measurement Behavior in Accordance with the SCOR Recommendation

The APEX Deep profiling float implements dissolved oxygen in-air measurements using Scientific Committee on Oceanic Research (SCOR) Working Group 142 recommendations as referenced in the note in the following page. When the float transitions to Surface phase, 10 measurements are taken at 15-second intervals before and after inflating the air bladder. The results are written to the science_log.txt file.

The SCOR sampling behavior is specified using a single entry in the sample.cfg file in accordance with the following syntax:

```
<SURFACE>
MEASURE OPT
```

NOTE *Bittig Henry, Kortzinger Arne, Johnson Ken, Claustre Hervé, Emerson Steve, Fennel Katja, Garcia Hernan, Gilbert Denis, Gruber Nicolas, Kang Dong-Jin, Naqvi Wajih, Prakash Satya, Riser Steven, Thierry Virginie, Tilbrook Bronte, Uchida Hiroshi, Ulloa Osvaldo, Xing Xiagang (2015). SCOR WG 142: Quality Control Procedures for Oxygen and Other Biogeochemical Sensors on Floats and Gliders. Recommendation for oxygen measurements from Argo floats, implementation of in-air-measurement routine to assure highest long-term accuracy. http://doi.org/10.13155/45917*

For example, to sample the oxygen sensor in accordance with the SCOR recommendation, enter:

```
<SURFACE>
MEASURE OPT
```

A.1.4 Reading the Science Log Oxygen Sensor Data

Oxygen sensor data are provided in the science_log.bin file. For instructions on how to uncompress this file and convert it to a CSV file, refer to “Converting the science_log.bin and vitals_log.bin Files to CSV Files” on page 4-19.

All rows in the science_log.csv file with the first column label "O2" contain oxygen sensor data. The columns are from left to right:

Sensor type:	O2
Time stamp:	UTC (YYYYMMDDTHHMMSS)
O2 concentration:	The absolute oxygen content in µM, where 1 Molar = 1 mole/liter
Air saturation:	The relative air saturation in % relative to the nominal air pressure (1013.25 hPa)
Temperature:	The ambient temperature in °C
Cal phase:	The calibrated phase in degrees
TC phase:	The temperature compensated phase in degrees
C1 RPh:	The phase measurement with blue excitation light in degrees

C2 RPh: The phase measurement with red excitation light in degrees

C1 amp: The amplitude measurement with blue excitation light in mV

C2 amp: The amplitude measurement with red excitation light in mV

Raw temp: The voltage from the thermistor bridge in mV

An example row of oxygen sensor data extracted from a science_log.csv file is shown in Figure A-2:

O2	20170303T045146	127.4541	35.67524	9.444273	42.32684	42.32684	49.97864	7.651794	1090.297	744.6091	397.0721

Figure A-2: Example Row of Oxygen Sensor Data in a science_log.csv File

For more information on interpreting the science_log.bin file along with additional examples, refer to "Converting the science_log.bin and vitals_log.bin Files to CSV Files" on page 4-19.

A.1.5 Show Command

The **Show** command displays all of the information available from the Aanderaa Oxygen Optode 4330 Oxygen Sensor. To use the **Show** command, a PC must be connected to the float and running a terminal program.

To display the optode configuration information, enter:

```
sys_show optode
```

The optode configuration will be displayed on the PC as in the following example:

```
Optode Product Name:             Optode Sensor
Optode Product Number:           4330
Optode Serial Number:            1521
Optode SW Version:               4.4.8
Optode SW ID:                    1940006
Optode Node Description:         Optode Sensor #1521
Optode FoilID:                   1206EM
Optode Enable Text:              Yes
Optode Enable Decimalformat:     No
Optode Enable Sleep:             Yes
Optode Enable Polled Mode:       No
Optode Enable SVUformula:        Yes
Optode Interval[s]:              3
```

```
Optode Analog TempLimit[Deg.C]:  -5.000000e+00 3.500000e+01
Optode Analog ConcLimit[uM]:     0.000000e+00 0.000000e+00
Optode Analog SatLimit:          0.000000e+00 0.000000e+00
Optode Analog PhaseLimit[Deg.C]: 0.000000e+00 0.000000e+00
Optode Analog Output:            1521
Optode Analog1 Coef:             0.000000e+00 0.000000e+00
Optode Analog2 Coef:             0.000000e+00 0.000000e+00
Optode Enable AirSaturation:     Yes
Optode Enable Rawdata:           Yes
Optode Enable Temperature:       Yes
Optode Enable HumidityComp:      Yes
Optode Salinity[PSU]:            0.000000e+00
Optode PhaseCoef:                0.000000e+00 1.000000e+00
0.000000e+00 0.000000e+00
Optode FoilCoefA[1-7]:           -2.988314e-06 -6.137785e-06
1.684659e-03 -1.857173e-01 6.784399e-04 -5.597908e-07
1.040158e+01
Optode FoilCoefA[8-14]:          -5.986907e-02 1.360425e-04 -
4.776977e-07 -3.032937e+02 2.530496e+00 -1.267045e-02
1.040454e-04
Optode FoilCoefB[1-7]:           -3.560390e-07 3.816713e+03 -
4.475507e+01 4.386162e-01 -7.146342e-03 8.906234e-05 -
6.343012e-07
Optode FoilCoefB[8-14]:          0.000000e+00 0.000000e+00
0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
0.000000e+00
Optode FoilPolyDegT[1-14]:       1 0 0 0 1 2 0 1 2 3 0 1 2 3
Optode FoilPolyDegT[15-28]:      4 0 1 2 3 4 5 0 0 0 0 0 0 0
Optode FoilPolyDegO[1-14]:       4 5 4 3 3 3 2 2 2 2 1 1 1 1
Optode FoilPolyDegO[15-28]:      1 0 0 0 0 0 0 0 0 0 0 0 0 0
Optode SVUFoilCoef:              2.933651e-03 1.245261e-04
2.482191e-06 2.317514e+02 -3.094964e-01 -5.569272e+01
4.559175e+00
Optode ConcCoef:                 0.000000e+00 1.000000e+00
Optode NomAirPress[hPa]:         1.013250e+03
Optode NomAirMix[hPa]:           0.000000e+00
Optode CalDataSat[Deg]:          3.174000e+01 9.910000e+00
Optode CalDataAPress[hPa]:       9.727329e+02
Optode CalDataZero[Deg]:         6.233000e+01 2.445500e+01
```

A.1.6 Other Commands

The following additional Aanderaa Oxygen Optode 4330 Oxygen Sensor commands are available:

optode_config. Runs the configuration sequence for the Aanderaa Oxygen Optode 4330 Oxygen Sensor.

Syntax: `optode_config`

optode_sample. Obtains sample data from the Aanderaa Oxygen Optode 4330 Oxygen Sensor.

Syntax: `optode_sample`

optode_test. Runs through configuration, displays configuration and obtains sample data from the Aanderaa Oxygen Optode 4330 Oxygen Sensor.

Syntax: `optode_test`

A.1.7 Oxygen Sensor (Aanderaa) Factory Configuration

The factory configuration of the installed Aanderaa Oxygen Optode 4330 Oxygen Sensor is the following:

```
Set Passkey(1000)
Set Salinity(0)
Set Comm TimeOut(1 min)
Set Mode(Smart Sensor Terminal)
Set Enable Text(Yes)
Set Flow Control(None)
Set Enable Comm Indicator(Yes)
Set Enable DecimalFormat(No)
Set Enable Temperature(Yes)
Set Enable AirSaturation(Yes)
Set Enable HumidityComp(Yes)
Set Enable RawData(Yes)
Set Enable Polled Mode(No)
Set Enable Sleep(Yes)
Set Enable RedReference(Yes)
Set Interval(3)
```

A.2 Oxygen Sensor (JFE Advantech)

The Teledyne Webb Research APEX Deep profiling float optionally includes the JFE Advantech RINKO FT Model ARO-FT Oxygen Sensor for periodic sampling of dissolved oxygen. The sensor is multipoint calibrated and based on the phosphorescence principle for fast response, high accuracy and long term stability in a compact, lightweight package.

A.2.1 Specifying the Oxygen Sensor Hardware Configuration

A line in the sensors.cfg file defines the oxygen sensor hardware configuration as in the following example:

```
jfe_rinko_ft COM5 B38400
```

For more information on the sensors.cfg file along with additional examples, refer to “sensors.cfg File” on page 4-16.

A.2.2 Specifying the Oxygen Sensor Sampling Behavior

The interval sampling behavior for the oxygen sensor is specified using a single entry in the sample.cfg file for each of any number of pressure ranges in accordance with the following syntax:

SAMPLE OPT *[<start> [<stop> [<interval> [<units> [<count>]]]]]*

For example, to sample the oxygen sensor every 10 dbar beginning at 1000 dbar and ending at 200 dbar, enter:

```
<ASCENT>
SAMPLE OPT 1000 200 10 DBAR
```

The start and stop pressures can be in any order. For more information on using the sample.cfg file to program the behavior of the float along with additional examples, refer to “sample.cfg File” on page 4-2.

A.2.3 Specifying the Oxygen Sensor In-Air Measurement Behavior in Accordance with the SCOR Recommendation

The APEX Deep profiling float implements dissolved oxygen in-air measurements using Scientific Committee on Oceanic Research (SCOR) Working Group 142 recommendations as referenced in the note in the following page. When the float transitions to Surface phase, 10 measurements are taken at 15-second intervals before and after inflating the air bladder. The results are written to the science_log.txt file.

The SCOR sampling behavior is specified using a single entry in the sample.cfg file in accordance with the following syntax:

```
<SURFACE>
MEASURE OPT
```

NOTE *Bittig Henry, Kortzinger Arne, Johnson Ken, Claustre Hervé, Emerson Steve, Fennel Katja, Garcia Hernan, Gilbert Denis, Gruber Nicolas, Kang Dong-Jin, Naqvi Wajih, Prakash Satya, Riser Steven, Thierry Virginie, Tilbrook Bronte, Uchida Hiroshi, Ulloa Osvaldo, Xing Xiagang (2015). SCOR WG 142: Quality Control Procedures for Oxygen and Other Biogeochemical Sensors on Floats and Gliders. Recommendation for oxygen measurements from Argo floats, implementation of in-air-measurement routine to assure highest long-term accuracy.* http://doi.org/10.13155/45917

For example, to sample the oxygen sensor in accordance with the SCOR recommendation, enter:

```
<SURFACE>
MEASURE OPT
```

A.2.4 Reading the Science Log Oxygen Sensor Data

Oxygen sensor data are provided in the science_log.bin file. For instructions on how to uncompress this file and convert it to a CSV file, refer to “Converting the science_log.bin and vitals_log.bin Files to CSV Files” on page 4-19.

All rows in the science_log.csv file with the first column label "O2" contain oxygen sensor data. The columns are from left to right:

Sensor type:	O2
Time stamp:	UTC (YYYYMMDDTHHMMSS)
Temperature:	The ambient temperature in A/D
Dissolved O2:	The dissolved oxygen in A/D
BL phase:	The blue Light phase in A/D
RL phase:	The red Light phase in A/D
BL amp:	The blue Light amplitude in A/D
RL amp:	The red Light amplitude in A/D
LED time:	The LED accumulated time in tens of milliseconds

An example row of oxygen sensor data extracted from a science_log.csv file is shown in Figure A-3:

O2	20170303T045146	34345	28422	38619	26898	9433	19622	53125

Figure A-3: Example Row of Oxygen Sensor Data in a science_log.csv File

For more information on interpreting the science_log.bin file along with additional examples, refer to “Converting the science_log.bin and vitals_log.bin Files to CSV Files” on page 4-19. In addition, refer to Chapter 8 of the JFE Advantech “RINKO FT User’s Manual” (p/n A-0133-04) for information on how to convert AD count values to scientific/engineering values.

A.2.5 Show Command

The **Show** command displays all of the information available from the JFE Advantech RINKO FT Model ARO-FT Oxygen Sensor. To use the **Show** command, a PC must be connected to the float and running a terminal program.

To display the optode configuration information, enter:

```
sys_show optode
```

The optode configuration will be displayed on the PC as in the following example:

```
Firmware Version : Ver.1.01
Serial Number    : 0000000006
C0               : 2.989382e-03
C1               : 1.249814e-04
C2               : 2.963225e-06
d0               : 2.910802e-04
d1               : -1.706489e-01
d2               : 1.798828e-01
d3               : 0.000000e+00
d4               : 0.000000e+00
Cp               : 0.000000e+00
e0               : 1.000000e+00
FilmNo           : 000000BA
docaldate        : 2014/01/01
A                : -1.310814e+01
B                : 1.487751e-03
C                : -2.710944e-08
D                : 6.412424e-13
E                : -8.228551e-18
```

```
F                  : 5.316122e-23
G                  : 0.000000e+00
H                  : 0.000000e+00
C0                 : 0.000000e+00
tcaldate           : 2017/03/18
```

A.2.6 Other Commands

The following additional JFE Advantech RINKO FT Model ARO-FT Oxygen Sensor commands are available:

optode_config. Runs the configuration sequence for the JFE Advantech RINKO FT Model ARO-FT Oxygen Sensor.

Syntax: `optode_config`

optode_sample. Obtains sample data from the JFE Advantech RINKO FT Model ARO-FT Oxygen Sensor.

Syntax: `optode_sample`

optode_test. Runs through configuration, displays configuration and obtains sample data from the JFE Advantech RINKO FT Model ARO-FT Oxygen Sensor.

Syntax: `optode_test`

optode_show. Displays all of the sensor information available from the JFE Advantech RINKO FT Model ARO-FT Oxygen Sensor.

Syntax: `optode_show`

A.2.7 Oxygen Sensor (JFE Advantech) Factory Configuration

The factory configuration of the installed JFE Advantech RINKO FT Model ARO-FT Oxygen Sensor is available from the manufacture.

A.3 Fluorometer

The Teledyne Webb Research APEX profiling float optionally includes one of three versions of the WET Labs Environmental Characterization Optics (ECO) Puck for periodic sampling of chlorophyll fluorescence and particulate backscattering: the FLBBAP2, the FLBBBB2K which includes a second backscattering wavelength, and the FLBBCDAP2 which includes a chromophoric dissolved organic matter (CDOM) channel.

Floats that include the fluorometer along with the CTD provide data essential for understanding oceanic biogeochemistry, the global carbon cycle and climate change. For example, the FLBB measures both chlorophyll fluorescence at 470 nm and backscattering at 700 nm within the same volume. Chlorophyll concentration is determined by directly measuring the amount of chlorophyll-a fluorescence emission from a given sample volume of water. Chlorophyll, when excited by the presence of an external light source, absorbs light in certain regions of the visible spectrum and re-emits a small portion of this light as fluorescence at longer wavelengths. The fluorometer is mounted to the float housing near the bottom. A cable from the sensor connects to a connector ring located at the bottom of the float housing.

A.3.1 Specifying the Fluorometer Hardware Configuration

A line in the sensors.cfg file defines the fluorometer hardware configuration.

For the FLBBAP2:

```
flbb COM5 B19200
```

For the FLBBBB2K:

```
flbb_bb COM5 B19200
```

For the FLBBCDAP2:

```
flbb_cd COM5 B19200
```

For more information on the sensors.cfg file along with additional examples, refer to “sensors.cfg File” on page 4-16.

A.3.2 Specifying the Fluorometer Sampling Behavior

The interval sampling behavior for *all three* versions of the fluorometer is specified using a single entry in the sample.cfg file for each of any number of pressure ranges in accordance with the following syntax:

SAMPLE FLBB *[<start> [<stop> [<interval> [<units> [<count>]]]]]*

For example, to sample the fluorometer every 10 dbar beginning at 1000 dbar and ending at 200 dbar, enter:

```
<ASCENT>
SAMPLE FLBB 1000 200 10 DBAR
```

The start and stop pressures can be in any order. For more information on using the sample.cfg file to program the behavior of the float along with additional examples, refer to “sample.cfg File” on page 4-2.

A.3.3 Reading the Science Log Fluorometer Data

Fluorometer data are provided in the science_log.bin file. For instructions on how to uncompress this file and convert it to a CSV file, refer to “Converting the science_log.bin and vitals_log.bin Files to CSV Files” on page 4-19.

All rows in the science_log.csv file with first column field starting with "FLBB" contain fluorometer data. For a fluorometer that includes a third channel, this channel, "BB" or "CD," is appended to the first field as in "FLBB_BB" and "FLBB_CD."

For the FLBBAP2 the columns are from left to right:

Sensor type:	FLBB
Time stamp:	UTC (YYYYMMDDTHHMMSS)
chl_wave:	The wavelength of the LED used for the fluorometer in nanometers
chl_sig:	The chlorophyll signature in micrograms per liter
bsc_wave:	The wavelength of the LED used for backscatter in nanometers
bsc_sig:	The backscatter in inverse meters per steradian
therm_sig:	The temperature signal

An example row of FLBBAP2 fluorometer data extracted from a science_log.csv file is shown in Figure A-4:

FLBB	20161103T031514	470	122	700	487	554

Figure A-4: Example Row of FLBBAP2 Fluorometer Data in a science_log.csv File

For the FLBBBB2K the columns are from left to right:

Sensor type:	FLBB_BB
Time stamp:	UTC (YYYYMMDDTHHMMSS)
chl_wave:	The wavelength of the LED used for the fluorometer in nanometers
chl_sig:	The chlorophyll signature in micrograms per liter
bsc_wave0:	First wavelength of the LED used for backscatter in nanometers
bsc_sig0:	First backscatter in inverse meters per steradain
bsc_wave1:	Second wavelength of the LED used for backscatter in nanometers
bsc_sig1:	Second backscatter in inverse meters per steradian
therm_sig:	The temperature signal

An example row of FLBBBB2K fluorometer data extracted from a science_log.csv file is shown in Figure A-5:

FLBB_BB	20161103T031514	695	582	700	4130	470	4130	541

Figure A-5: Example Row of FLBBBB2K Fluorometer Data in a science_log.csv File

For the FLBBCDAP2 the columns are from left to right:

Sensor type:	FLBB_CD
Time stamp:	UTC (YYYYMMDDTHHMMSS)
chl_wave:	The wavelength of the LED used for the fluorometer in nanometers
chl_sig:	The chlorophyll signature in micrograms per liter
bsc_wave:	The wavelength of the LED used for backscatter in nanometers
bsc_sig:	The backscatter in inverse meters per steradain.
cd_wave:	The wavelength of the LED used for CDOM in nanometers
cd_sig:	The CDOM in ppb
therm_sig:	The temperature signal

An example row of FLBBCDAP2 fluorometer data extracted from a science_log.csv file is shown in Figure A-6:

FLBB_CD	20160319T033732	695	582	700	4130	470	4130	541

Figure A-6: Example Row of FLBBCDAP2 Fluorometer Data in a science_log.csv File

For more information on interpreting the science_log.bin file along with additional examples, refer to “Converting the science_log.bin and vitals_log.bin Files to CSV Files” on page 4-19.

A.3.4 Show Command

The **Show** command displays all of the information available from the WET Labs ECO Puck Fluorometer. To use the **Show** command, a PC must be connected to the float and running a terminal program.

To display the fluorometer configuration information, enter:

```
sys_show flbb
```

For an FLBB the configuration will be displayed on the PC as in the following example:

```
FLBB Serial Number:  FLBBAP2-1578
FLBB Version:        FLNTUAPX 4.04
FLBB Average:        Set in Factory
FLBB Packets:        0
```

For an FLBBBB2K the configuration will be displayed on the PC as in the following example:

```
FLBB_BB Serial Number:  FLBBBB2K-4240
FLBB_BB Version:        TripletD 4.07
FLBB_BB Average:        Set in Factory
FLBB_BB Packets:        0
```

For an FLBB_CD the configuration will be displayed on the PC as in the following example:

```
FLBB_CD Serial Number:  FLBBCDAP2-3210
FLBB_CD Version:        FLBBWAP2 v4.07
FLBB_CD Average:        Set in Factory
FLBB_CD Packets:        0
```

A.3.5 Other Commands

The following additional WET Labs ECO Puck Fluorometer commands are available:

flbb_config. Runs the configuration sequence for the WET Labs ECO Puck Fluorometer.

Syntax: `flbb_config`

flbb_sample. Obtains sample data from the WET Labs ECO Puck Fluorometer.

Syntax: `flbb_sample`

flbb_test. Runs through configuration, displays configuration and obtains sample data from the WET Labs ECO Puck Fluorometer.

Syntax: `flbb_test`

flbb_show. Displays all of the sensor information available from the WET Labs ECO Puck Fluorometer.

Syntax: `flbb_show`

A.3.6 Fluorometer Factory Configuration

The factory configuration of the installed WET Labs (ECO) Puck is the following:

```
$pkt=0
$sto
```

A.4 Transmissometer

The Teledyne Webb Research APEX profiling float optionally includes the WET Labs C-Rover Transmissometer for periodic sampling of optical transmittance of sea water. The transmissometer is mounted to the float housing near the bottom. A cable from the sensor connects to a connector ring located at the bottom of the float housing.

A.4.1 Specifying the Transmissometer Configuration

A line in the sensors.cfg file defines the transmissometer hardware configuration:

```
crover COM11 B19200
```

For more information on the sensors.cfg file along with additional examples, refer to “sensors.cfg File” on page 4-16.

A.4.2 Specifying the Transmissometer Sampling Behavior

The interval sampling behavior for the C-Rover Transmissometer is specified using a single entry in the sample.cfg file for each of any number of pressure ranges in accordance with the following syntax:

SAMPLE XMIS *[<start> [<stop> [<interval> [<units> [<count>]]]]]*

For example, to sample the transmissometer every 10 dbar beginning at 1000 dbar and ending at 200 dbar, enter:

```
<ASCENT>
SAMPLE XMIS 1000 200 10 DBAR
```

To sample the transmissometer during the Park phase, enter:

```
<PARK>
SAMPLE XMIS
```

ParkTimerInterval determines the sample rate at the Park depth. For more information on using the sample.cfg file to program the behavior of the float along with additional examples, refer to “sample.cfg File” on page 4-2.

A.4.3 Reading the Science Log Transmissometer Data

Transmissometer data are provided in the science_log.bin file. For instructions on how to uncompress this file and convert it to a CSV file, refer to “Converting the science_log.bin and vitals_log.bin Files to CSV Files” on page 4-19.

All rows in the science_log.csv file with the first column label "CROVER" contain transmissometer data. The columns are from left to right:

Sensor type:	CROVER
Time stamp:	UTC (YYYYMMDDTHHMMSS)
reference:	Reference raw count
raw_sig:	Signal raw count
corr_sig:	Signal corrected count
therm:	The thermistor temperature raw count
attenuation:	Calculated beam attenuation in inverse meters

An example row of transmissometer data extracted from a science_log.csv file is shown in Figure A-7:

CROVER	20160319T033739	12181	7	7	544	30.8

Figure A-7: Example Row of Transmissometer Data in a science_log.csv File

For more information on interpreting the science_log.bin file along with additional examples, refer to “Converting the science_log.bin and vitals_log.bin Files to CSV Files” on page 4-19.

A.4.4 Show Command

The **Show** command displays all of the information available from the WET Labs C-Rover Transmissometer. To use the **Show** command, a PC must be connected to the float and running a terminal program.

To display the transmissometer configuration information, enter:

```
sys_show xmis
```

The tranmissometer configuration will be displayed on the PC as in the following example:

```
CROVER Model/Type:      BAM
CROVER Serial Number:   CRV2K-079
CROVER Version:         CSTAR 4.14
CROVER Average:         30
CROVER Packets:         0
```

A.4.5 Transmissometer Factory Configuration

The factory configuration of the installed WET Labs C-Rover Transmissometer is the following:

```
$ave=30
$pkt=0
$sto
```

A.5 Radiance Radiometer

The Teledyne Webb Research APEX profiling float optionally includes the Satlantic OCR-504 R10W Radiance Radiometer which combines precision optics and high performance microelectronics for periodic sampling of spectral light in an ocean environment. The radiance radiometer is externally mounted to the float near the bottom. A cable from the sensor connects to a connector ring located at the bottom of the float housing.

A.5.1 Specifying the Radiance Radiometer Hardware Configuration

A line in the sensors.cfg file defines the radiance radiometer hardware configuration:

```
satlantic_504r COM9 B9600
```

For more information on the sensors.cfg file along with additional examples, refer to "sensors.cfg File" on page 4-16.

A.5.2 Specifying the Radiance Radiometer Sampling Behavior

The interval sampling behavior for the radiance radiometer is specified using a single entry in the sample.cfg file for each of any number of pressure ranges in accordance with the following syntax:

SAMPLE RAD *[<start> [<stop> [<interval> [<units> [<count>]]]]]*

For example, to sample the radiance radiometer every 10 dbar beginning at 1000 dbar and ending at 200 dbar, enter:

```
<ASCENT>
SAMPLE RAD 1000 200 10 DBAR
```

The start and stop pressures can be in any order. For more information on using the sample.cfg file to program the behavior of the float along with additional examples, refer to "sample.cfg File" on page 4-2.

A.5.3 Reading the Science Log Radiance Radiometer Data

Radiance radiometer data are provided in the science_log.bin file. For instruction on how to uncompress this file and convert it to a CSV file, refer to "Converting the science_log.bin and vitals_log.bin Files to CSV Files" on page 4-19.

All rows in the science_log.csv file with the first column label "504R" contain irradiance radiometer data. The columns are from left to right:

Sensor type:	504R
Time stamp:	UTC (YYYYMMDDTHHMMSS)

Channel1: An AF formatted value representing the calibrated output in $uW/cm^2/nm$ from channel 1

Channel2: An AF formatted value representing the calibrated output in $uW/cm^2/nm$ from channel 2

Channel3: An AF formatted value representing the calibrated output in $uW/cm^2/nm$ from channel 3

Channel4: An AF formatted value representing the calibrated output in $uW/cm^2/nm$ from channel 4

An example row of radiance radiometer data extracted from a science_log.csv file is shown in Figure A-8:

504R	20160323T204807	0.003	0.007	0.0158	0.0151

Figure A-8: Example Row of Radiance Radiometer Data in a science_log.csv File

For more information on interpreting the science_log.bin file along with additional examples, refer to "Converting the science_log.bin and vitals_log.bin Files to CSV Files" on page 4-19.

A.5.4 Show Command

The **Show** command displays all of the information available from the Satlantic OCR-504 R10W Radiance Radiometer. To use the **Show** command, a PC must be connected to the float and running a terminal program.

To display the radiance radiometer configuration information, enter:

```
sys_show rad
```

The radiance radiometer configuration will be displayed on the PC as in the following example:

```
Satlantic OCR-504 Multispectral Radiometer
Firmware version: 5.1.0 - SatNet Type B
Sensor Type:                      radiance
Serial Number:                    0022
Frame Type:                       short
Header:                           SATFR4
Telemetry Baud Rate:              9600 bps
Maximum Frame Rate:               1 Hz
Initialize Silent Mode:           on
Initialize Power Down:            off
Initialize Automatic Telemetry:   on
Network Mode:                     off
```

```
networkAddress:                 100
Network Baud Rate:              38400 bps
Averaging:                      on
Immersed:                       on
Calibrated Output:              on
Sensor Latency:                 0
Calibration Coefficients:
    Optical Channel 1:
        a0: 2148023214.5
        a1: 2.653745e-09
        im: 1.758

    Optical Channel 2:
        a0: 2147769455.6
        a1: 2.538758e-09
        im: 1.752

    Optical Channel 3:
        a0: 2147559434.8
        a1: 2.723878e-09
        im: 1.746

    Optical Channel 4:
        a0: 2148019892.9
        a1: 1.587890e-09
        im: 1.739
```

A.5.5 Other Commands

The following additional Satlantic OCR-504 R10W Radiance Radiometer commands are available:

rad_config. Runs the configuration sequence for the Satlantic OCR-504 R10W Radiance Radiometer.

Syntax: `rad_config`

rad_sample. Obtains sample data from the Satlantic OCR-504 R10W Radiance Radiometer.

Syntax: `rad_sample`

rad_test. Runs through configuration, displays configuration and obtains sample data from the Satlantic OCR-504 R10W Radiance Radiometer.

Syntax: `rad_test`

rad_show. Displays all of the sensor information available from the Satlantic OCR-504 R10W Radiance Radiometer.

Syntax: `rad_show`

A.5.6 Radiance Radiometer Factory Configuration

The factory configuration of the installed Satlantic OCR-504 R10W Radiance Radiometer is the following:

```
set frametype short
set initsm on
set maxrate 1
set initpd off
set initat on
set avg on
set immersed on
set usecal on
set latency 0
set netmode off
```

A.6 Irradiance Radiometer (Satlantic)

The Teledyne Webb Research APEX profiling float optionally includes the Satlantic OCR-504 ICSW Irradiance Radiometer which combines precision optics and high performance microelectronics for periodic sampling of spectral light in an ocean environment. The irradiance radiometer is externally mounted to the float near the top. A cable from the sensor connects to a connector ring located at the top of the float housing.

A.6.1 Specifying the Irradiance Radiometer Hardware Configuration

A line in the sensors.cfg file defines the irradiance radiometer hardware configuration:

```
satlantic_504i COM10 B9600
```

For more information on the sensors.cfg file along with additional examples, refer to “sensors.cfg File” on page 4-16.

A.6.2 Specifying the Irradiance Radiometer Sampling Behavior

The interval sampling behavior for the irradiance radiometer is specified using a single entry in the sample.cfg file for each of any number of pressure ranges in accordance with the following syntax:

SAMPLE IRAD *[<start> [<stop> [<interval> [<units> [<count>]]]]]*

For example, to sample the irradiance radiometer every 10 dbar beginning at 1000 dbar and ending at 200 dbar, enter:

```
<ASCENT>
SAMPLE IRAD 1000 200 10 DBAR
```

The start and stop pressures can be in any order. For more information on using the sample.cfg file to program the behavior of the float along with additional examples, refer to “sample.cfg File” on page 4-2.

A.6.3 Reading the Science Log Irradiance Radiometer Data

Irradiance radiometer data are provided in the science_log.bin file. For instructions on how to uncompress this file and convert it to a CSV file, refer to “Converting the science_log.bin and vitals_log.bin Files to CSV Files” on page 4-19.

All rows in the science_log.csv file with the first column label "504I" contain irradiance radiometer data. The columns are from left to right:

Sensor type:	504I
Time stamp:	UTC (YYYYMMDDTHHMMSS)

Channel1: An AF formatted value representing the calibrated output in uW/cm2/nm from channel 1

Channel2: An AF formatted value representing the calibrated output in uW/cm2/nm from channel 2

Channel3: An AF formatted value representing the calibrated output in uW/cm2/nm from channel 3

Channel4: An AF formatted value representing the calibrated output in uW/cm2/nm from channel 4

An example row of irradiance radiometer data extracted from a science_log.csv file is shown in Figure A-9:

504I	20160323T204743	0.1369	0.3072	0.0223	0.0479

Figure A-9: Example Row of Irradiance Radiometer Data in a science_log.csv File

For more information on interpreting the science_log.bin file along with additional examples, refer to “Converting the science_log.bin and vitals_log.bin Files to CSV Files” on page 4-19.

A.6.4 Show Command

The **Show** command displays all of the information available from the Satlantic OCR-504 ICSW Irradiance Radiometer. To use the **Show** command, a PC must be connected to the float and running a terminal program.

To display the irradiance radiometer configuration information, enter:

```
sys_show irrad
```

The irradiance radiometer configuration will be displayed on the PC as in the following example:

```
Satlantic OCR-504 Multispectral Radiometer
Firmware version: 5.1.0 - SatNet Type B
Sensor Type:                      irradiance
Serial Number:                    0400
Frame Type:                       short
Header:                           SATFI4
Telemetry Baud Rate:              9600 bps
Maximum Frame Rate:               1 Hz
Initialize Silent Mode:           on
Initialize Power Down:            off
Initialize Automatic Telemetry:   on
```

```
Network Mode:                    off
networkAddress:                  200
Network Baud Rate:               38400 bps
Averaging:                       on
Immersed:                        on
Calibrated Output:               on
Sensor Latency:                  0
Calibration Coefficients:
    Optical Channel 1:
        a0: 2147551528.8
        a1: 2.141215e-07
        im: 1.368

    Optical Channel 2:
        a0: 2147318656.7
        a1: 2.071143e-07
        im: 1.410

    Optical Channel 3:
        a0: 2147887045.0
        a1: 2.060970e-07
        im: 1.365

    Optical Channel 4:
        a0: 2147172741.4
        a1: 2.039716e-07
        im: 1.372
```

A.6.5 Other Commands

The following additional Satlantic OCR-504 ICSW Irradiance Radiometer commands are available:

irad_config. Runs the configuration sequence for the Satlantic OCR-504 ICSW Irradiance Radiometer.

Syntax: `irad_config`

irad_sample. Obtains sample data from the Satlantic OCR-504 ICSW Irradiance Radiometer.

Syntax: `irad_sample`

irad_test. Runs through configuration, displays configuration and obtains sample data from the Satlantic OCR-504 ICSW Irradiance Radiometer.

Syntax: `irad_test`

irad_show. Displays all of the sensor information available from the Satlantic OCR-504 ICSW Irradiance Radiometer.

Syntax: `irad_show`

A.6.6 Irradiance Radiometer Factory Configuration

The factory configuration of the installed Satlantic OCR-504 ICSW Irradiance Radiometer is the following:

```
set frametype short
set initsm on
set maxrate 1
set initpd off
set initat on
set avg on
set immersed on
set usecal on
set latency 0
set netmode of
```

A.7 Irradiance Radiometer (TriOS)

The Teledyne Webb Research APEX profiling float optionally includes the TriOS RAMSES G2 ACC Irradiance Radiometer. The irradiance radiometer is externally mounted to the float near the top. A cable from the sensor connects to a connector ring located at the top of the float housing

A.7.1 Specifying the Irradiance Radiometer Hardware Configuration

A line in the sensors.cfg file defines the irradiance radiometer hardware configuration:

```
sensor trios_ramses COM4 B9600
```

For more information on the sensors.cfg file along with additional examples, refer to “sensors.cfg File” on page 4-16.

A.7.2 Specifying the Irradiance Radiometer Powering Behavior

The TriOS RAMSES G2 ACC Irradiance Radiometer may take up to 60 seconds to power on. When it is time to sample, the irradiance radiometer will be automatically powered on for the sample. However, it can be manually powered on and manually powered off to avoid a warmup time when sampling less than 60 seconds.

The power behavior for the irradiance radiometer is specified using a single entry in the sample.cfg file for each of any number of pressure ranges in accordance with the following syntax:

Power IRAD *[<start> [<stop>]]*

For example, to power the irradiance radiometer beginning at 200 dbar and ending at 50 dbar, enter:

```
<ASCENT>
Power IRAD 200 50
```

A.7.3 Specifying the Irradiance Radiometer Sampling Behavior

The interval sampling behavior for the irradiance radiometer is specified using a single entry in the sample.cfg file for each of any number of pressure ranges in accordance with the following syntax:

Sample IRAD *[<start> [<stop> [<interval> [<units> [<count>]]]]]*

For example, to sample the irradiance radiometer every 5 dbar beginning at 200 dbar and ending at 50 dbar, enter:

```
<ASCENT>
Sample IRAD 200 50 5 dbar
```

The start and stop pressures can be in any order. For more information on using the sample.cfg file to program the behavior of the float along with additional examples, refer to “sample.cfg File” on page 4-2.

A.7.4 Reading the Science Log Irradiance Radiometer Data

Irradiance radiometer data are provided in the science_log.bin file. For instructions on how to uncompress this file and convert it to a CSV file, refer to “Converting the science_log.bin and vitals_log.bin Files to CSV Files” on page 4-19.

All rows in the science_log.csv file with the first column label "IRAD" contain irradiance radiometer data. The columns are from left to right:

Sensor type:	IRAD
Time stamp:	UTC (YYYYMMDDTHHMMSS)
Integration Time:	The integration time of the spectrometer used for the measurement
Temperature:	The temperature during the last measurement in °C as taken from the pressure sensor
Pressure:	The pressure during last measurement in bars
Pre Inclination:	The inclination angle in degrees (0-360) taken before the light measurement
Post Inclination:	The inclination angle in degrees (0-360) taken after the light measurement

An example row of irradiance radiometer data extracted from a science_log.csv file is shown in Figure A-10.

IRAD	20200215T032019	4096	22.16563	0.95728	270.3516	270.379

Figure A-10: Example Row of Irradiance Radiometer Data in a science_log.csv File

For more information on interpreting the science_log.bin file along with additional examples, refer to “Converting the science_log.bin and vitals_log.bin Files to CSV Files” on page 4-19.

A.7.5 Reading the Irradiance Log Raw Ordinate Data

The irad_log CSV file contains 255 samples of values of the ordinate of the graph of the last measured spectrum as raw values between 0 and 65535.

A.7.6 Show Command

The **Show** command displays all of the information available from the TriOS RAMSES G2 ACC Irradiance Radiometer. To use the **Show** command, a PC must be connected to the float and running a terminal program.

To display the irradiance radiometer configuration information, enter:

```
sys_show irrad
```

The irradiance radiometer configuration will be displayed on the PC as in the following example:

```
Serial Number:        01600004
Firmware Version:     1.0.4
Self Trig Enable:     0
Self Trig Interval:   15
Integration Time:     0
LAN Enable:           0x0
DateTime:             1582311943
Device Description:
Data Comment 1:
Data Comment 2:
Data Comment 3:
Data Comment 4:
```

A.7.7 Other Commands

The following additional TriOS RAMSES G2 ACC Irradiance Radiometer commands are available:

irad_config. Runs the configuration sequence for the TriOS RAMSES G2 ACC Irradiance Radiometer.

Syntax: `irad_config`

irad_sample. Obtains sample data from the TriOS RAMSES G2 ACC Irradiance Radiometer.

Syntax: `irad_sample`

irad_test. Runs through configuration, displays configuration and obtains sample data from the TriOS RAMSES G2 ACC Irradiance Radiometer.

Syntax: `irad_test`

irad_show. Displays all of the sensor information available from the TriOS RAMSES G2 ACC Irradiance Radiometer.

Syntax: `irad_show`

A.7.8 Irradiance Radiometer Factory Configuration

The factory configuration of the installed TriOS RAMSES G2 ACC Irradiance Radiometer is the following:

```
set self trig enable 0
set integration time 0
set LAN ENABLE 0x0000 (OFF)
```

A.8 Nitrate Sensor

The Teledyne Webb Research APEX profiling float optionally includes the Sea-Bird Scientific Satlantic Deep SUNA (Submersible Ultraviolet Nitrate Analyzer) V2 Nitrate Sensor for measuring nitrate concentrations in the ocean based on the absorption characteristics of nitrate in the UV light spectrum.

A.8.1 Specifying the Nitrate Sensor Hardware Configuration

A line in the sensors.cfg file defines the nitrate sensor hardware configuration:

```
satlantic_suna COM11 B9600
```

For more information on the sensors.cfg file along with additional examples, refer to “sensors.cfg File” on page 4-16.

A.8.2 Specifying the Nitrate Sensor Sampling Behavior

The interval sampling behavior for the nitrate sensor is specified using a single entry in the sample.cfg file for each of any number of pressure ranges in accordance with the following syntax:

SAMPLE SUNA *[<start> [<stop> [<interval> [<units> [<count>]]]]]*

For example, to sample the nitrate sensor every 5 dbar beginning at 500 dbar and ending at 5 dbar, enter:

```
<ASCENT>
SAMPLE SUNA 500 5 5 DBAR
```

The start and stop pressures can be in any order. For more information on using the sample.cfg file to program the behavior of the float along with additional examples, refer to “sample.cfg File” on page 4-2.

NOTE *In 2009 Sakamoto developed an algorithm to calculate the most accurate nitrate concentrations by recalculating sea salt extinction using the sample temperature, subtracting the expected absorbance due to the CTD measured salinity at that temperature, referred to as temperature corrected, salinity subtracted (TCSS). Implementing the following sample configuration ensures that the float samples the CTD immediately prior to nitrate sampling and provides the pressure, temperature and salinity to the nitrate sensor to enable TCSS nitrate calculation in real time.*

For example, to sample the CTD every 5 dbar beginning at 1000 dbar and ending at 5 dbar and the nitrate sensor every 5 dbar beginning at 500 dbar and ending at 5 dbar, enter:

```
<ASCENT>
SAMPLE CTD 1000 5 5 DBAR
SAMPLE SUNA 500 5 5 DBAR
```

NOTE *When sampling with a pumped type CTD and a specified **SurfacePressure** (typically 4 dbar), the nitrate sensor will complete the sampling at the specified **SurfacePressure**. For a non-pumped type CTD, the **SurfacePressure** can be set to 0 for interval sampling with the nitrate sensor up to the surface.*

For example, to sample a pumped type CTD continuously beginning at 1500 dbar and ending at the pressure specified by **SurfacePressure**, enter:

```
<ASCENT>
PROFILE CTD 1500 0
```

A.8.3 Reading the Science Log Nitrate Sensor Data

Nitrate sensor data are provided in the science_log.bin file. For instructions on how to uncompress this file and convert it to a CSV file, refer to “Converting the science_log.bin and vitals_log.bin Files to CSV Files” on page 4-19.

All rows in the science_log.csv file with the first column label "NO3" contain nitrate sensor data. The columns are from left to right:

Sensor type:	NO3
Time stamp:	UTC (YYYYMMDDTHHMMSS)
Nitrate:	Micromolar units (µM)

An example row of nitrate sensor data extracted from a science_log.csv file is shown in Figure A-11.

NO3	20170712T174004	0.04

Figure A-11: Example Row of Nitrate Sensor Data in a science_log.csv File

For more information on interpreting the science_log.bin file along with additional examples, refer to “Converting the science_log.bin and vitals_log.bin Files to CSV Files” on page 4-19.

A.8.4 Reading the Nitrate Sensor Frame Data

The suna_log.txt file contains nitrate APF frames associated with each nitrate sample in the science_log.bin file. For the APF frame field formats, refer to the current version of the Sea-Bird Scientific Satlantic "Deep SUNA V2 Manual." To convert the suna_log.txt file to readable text, enter `gunzip suna_log.txt.gz`. The file will be unzipped to the suna_log.txt file which can be opened in a text editor.

The *actual* file name is prepended with the username, the profile number, the timestamp, and the file type as follows:

Syntax: *<username>.<profile number>.<timestamp>.<filename>.<filetype>.<ext>*

username:	The account username included in system.cfg
profile number:	A zero-padded 3-decimal number indicating the profile number. For example, the first descent would have a value of 001.
timestamp:	Date and time of the file creation in YYYYMMDDTHHMMSS format
filename:	Name of file
filetype:	Type of log file (.txt)
ext:	gz

For example, in the following file the username is f0036 and the profile number is 003:

f0036.003.20160122T193806.suna_log.txt.gz

A.8.5 Show Command

The **Show** command displays all of the information available from the Sea-Bird Scientific Satlantic Deep SUNA V2 Nitrate Sensor. To use the **Show** command, a PC must be connected to the float and running a terminal program.

To display the nitrate sensor configuration information, enter:

```
sys_show suna
```

The nitrate sensor configuration will be displayed on the PC as in the following example:

```
SUNA|Serial Number           : SN:1024
SUNA|Application Name        : SUNA-V2
SUNA|Firmware Version        : 2.2.14
SUNA|App Build Date          : Aug 30 2013, 10:43:21
SUNA|Operating Mode          : APF
SUNA|Power Up Time (UTC)     : 08/29/2017 14:16:06
SUNA|Current Time (UTC)      : 08/29/2017 14:16:07
SUNA|Power Active Timeout    : 10
SUNA|Power Cycle Counter     : 515
SUNA|System Reset Counter    : 522
SUNA|Sample Record Counter   : 700
```

```
SUNA|Last Sample Rec. Time  : 08/28/2017 21:08:49
SUNA|Error Record Counter   : 6
SUNA|Last Error Event Type  : -6
SUNA|Last Error Event Time  : 08/11/2017 13:22:26
SUNA|Fiberlite Odometer     : 0050:03:36
SUNA|Calibration File       : SNA1024A.CAL
SUNA|Spec Integration Time  : 225
SUNA|Wavelength Fit Range   : 217.00 <-> 240.00
SUNA|Fitted Concentrations  : 3
SUNA|Baseline Model         : 1
SUNA|Br Temp Compensation   : On
SUNA|Bromide Term in Model  : will be FIXED (to external CTD value)
SUNA|Absorbance Cutoff      : 10.000000
```

A.8.6 Other Commands

The following additional Sea-Bird Scientific Satlantic Deep SUNA V2 Nitrate Sensor commands are available:

suna_bake. Runs the deployment cleaning sequence for the Sea-Bird Scientific Satlantic Deep SUNA V2 Nitrate Sensor.

Syntax: `suna_bake`

suna_config. Runs the configuration sequence for the Sea-Bird Scientific Satlantic Deep SUNA V2 Nitrate Sensor.

Syntax: `suna_config`

suna_sample. Obtains sample data from the Sea-Bird Scientific Satlantic Deep SUNA V2 Nitrate Sensor.

Syntax: `suna_sample`

suna_show. Displays all of the sensor information available from the Sea-Bird Scientific Satlantic Deep SUNA V2 Nitrate Sensor.

Syntax: `suna_show`

suna_test. Runs through configuration, displays configuration and obtains sample data from the Sea-Bird Scientific Satlantic Deep SUNA V2 Nitrate Sensor.

Syntax: `suna_test`

A.8.7 Nitrate Sensor Factory Configuration

The factory configuration of the installed Sea-Bird Scientific Satlantic Deep SUNA V2 Nitrate Sensor is available from the manufacturer. In addition, Teledyne Webb Research provides the following updates to the factory configuration.

- RELAYBRD Missing (Disable Relay Board option if configured). This configuration informs the nitrate sensor that the optional Relay board is not installed.
- SKPSLEEP On (Float manages the nitrate sensor power). This configuration allows the float to manage the powering on and off of the nitrate sensor.
- LGTAVERS 5 (Average 5 fast sequence samples for noise reduction). This configuration instructs the nitrate sensor to take 5 fast sequence samples and return the average, helping to reduce sampling noise.

A.8.8 Nitrate Sensor Cleaning

During deployment the nitrate sensor can be cleaned periodically during the Park Descent, Park, or Deep Descent phase. The cleaning is specified by entering the phase followed by a single line entry in the sample.cfg file.

Syntax: *<phase>* Bake SUNA *[<duration> [<cycle> [<count>]]]*

phase:	PARKDESCENT PARK DEEPDESCENT
duration:	Duration for sensor bake (1–15) seconds
cycle:	Profiling cycle repetition (1–50)
count:	Repetition counter (1–10)

For a sample.cfg file entry that *does not* include one or more of the optional parameters, the float uses the following default entries:

duration:	15
cycle:	20
count:	1

Example. To clean the nitrate sensor during the Park Descent phase 6 times for 15 seconds each every 30 profiles, enter:

```
<PARKDESCENT>
Bake SUNA 15 30 6
```

A.9 Compass

The Teledyne Webb Research APEX profiling float optionally includes the True North Technologies (TNT) Electronic Compass for periodic sampling of heading and tilt using a 3-axis magnetometer and a 2-axis tilt sensor. The compass is internally mounted and connected.

A.9.1 Specifying the Compass Hardware Configuration

A line in the sensors.cfg file defines the compass hardware configuration:

```
tnt_compass COM5 B9600
```

For more information on the sensors.cfg file along with additional examples, refer to “sensors.cfg File” on page 4-16.

A.9.2 Specifying the Compass Sampling Behavior

The interval sampling behavior for the compass is specified using a single entry in the sample.cfg file for each of any number of pressure ranges in accordance with the following syntax:

SAMPLE CMP *[<start> [<stop> [<interval> [<units> [<count>]]]]]*

For example, to sample the compass every 10 dbar beginning at 1000 dbar and ending at 200 dbar, enter:

```
<ASCENT>
SAMPLE CMP 1000 200 10 DBAR
```

NOTE *Interval sampling for the compass should coincide with that of the transmissometer, the radiometer and the irradiometer if one or more of these sensors are installed.*

The start and stop pressure can be in any order. For more information on using the sample.cfg file to program the behavior of the float along with additional examples, refer to “sample.cfg File” on page 4-2.

A.9.3 Reading the Science Log Compass Data

Compass data are provided in the science_log.bin file. For instructions on how to uncompress this file and convert it to a CSV file, refer to “Converting the science_log.bin and vitals_log.bin Files to CSV Files” on page 4-19.

All rows in the science_log.csv file with the first column label "Compass" contain compass data. The columns are from left to right:

Sensor type:	Compass
Time stamp:	UTC (YYYYMMDDTHHMMSS)
Heading:	The float heading in degrees from magnetic north
Pitch:	The float pitch in degrees from horizontal in the direction of motion/heading
Roll:	The float roll in degrees from horizontal, perpendicular to the direction of motion/heading
Dip:	The float dip in degrees with respect to horizontal

An example row of compass data extracted from a science_log.csv file is shown in Figure A-12:

Compass	20131103T044940	231.8000031	-0.100000001	-2.700000048	2903

Figure A-12: Example Row of Compass Data in a science_log.csv File

For more information on interpreting the science_log.bin file along with additional examples, refer to “Converting the science_log.bin and vitals_log.bin Files to CSV Files” on page 4-19.

A.9.4 Show Command

The **Show** command displays all of the information available from the True North Technologies (TNT) Electronic Compass. To use the **Show** command, a PC must be connected to the float and running a terminal program.

To display the compass configuration information, enter:

```
sys_show compass
```

The compass configuration will be displayed on the PC as in the following example:

```
TNT Device Id:          357048
TNT Firmware Revision: TNT1501 Rev 2.64 - PCB 1511 Rev E - 08/06/07
```

A.9.5 Other Commands

The following additional True North Technologies (TNT) Electronic Compass commands are available:

tnt_config. Runs the configuration sequence for the True North Technologies (TNT) Electronic Compass.

Syntax: `tnt_config`

tnt_heading. Obtains data from the True North Technologies (TNT) Electronic Compass.

Syntax: `tnt_heading`

tnt_show. Displays all of the sensor information available from the True North Technologies (TNT) Electronic Compass.

Syntax: `tnt_show`

tnt_test. Runs through configuration, displays configuration and obtains sample data from the True North Technologies (TNT) Electronic Compass.

Syntax: `tnt_test`

A.9.6 Compass Factory Configuration

The factory configuration of the installed True North Technologies (TNT) Electronic Compass is as provided by the manufacturer.

A.10 Seascan RAFOS Sensor

The Teledyne Webb Research APEX profiling float optionally includes the Seascan RAFOS digital acoustic receiver module which receives acoustic signals to position floats in sub-surface drift applications, such as under ice. The sensor is designed to listen for a designated period at a specific time of day each day during the Park phase and report time indexed correlation magnitudes. The module is internally mounted with an external hydrophone.

A.10.1 Specifying the RAFOS Sensor Hardware Configuration

A line in the sensors.cfg file defines the RAFOS sensor hardware configuration:

```
sensor seascan_rafos COM4 B9600
```

For more information on the sensors.cfg file along with additional examples, refer to “sensors.cfg File” on page 4-16.

A.10.2 Specifying the RAFOS Sensor Listening Behavior

The interval listening behavior for the RAFOS sensor is specified using a single entry in the sample.cfg file in accordance with the following syntax:

LISTEN RAFOS *[start_day_time [duration]]*

start_day_time:	The minute of every day from 12:00 AM in which the RAFOS sensor will turn on to listen (0–1439), default (0)
duration:	The time in minutes of the listening window (0–335 or 0.3075 sec/sample with max 65535 samples), default (120)

For example, to start listening at a time of day of 225 minutes from 12:00 AM (3:45 PM), for a duration of 5 minutes, enter:

```
<PARK>
LISTEN RAFOS 225 5
```

NOTE *The RAFOS sensor does not support spot sampling.*

NOTE *If a Seascan RTC is included in the sensors.cfg, the RAFOS sensor listening specification will automatically capture the RTC’s timestamp prior to the start of the listening window. Additionally a PTS sample will occur automatically after the RAFOS sensor listening window.*

NOTE *Because the listening window is based on the Seascan RTC which can drift approximately 1 minute per year, it is recommended to start and end the listening window a few minutes before and after the exact window desired.*

NOTE *Listening will start if the Park phase starts during a listening window and listening will end if the Park phase ends during a listening window.*

For more information on using the sample.cfg file to program the behavior of the float along with additional examples, refer to "sample.cfg File" on page 4-2.

A.10.3 Reading the Science Log RAFOS Sensor Data

RAFOS sensor data are provided in the science_log.bin file. For instructions on how to uncompress this file and convert it to a CSV file, refer to "Converting the science_log.bin and vitals_log.bin Files to CSV Files" on page 4-19.

All rows in the science_log.csv file with the first column label "RAFOS" contain RAFOS sensor data. The columns are from left to right:

Sensor type: RAFOS

Time stamp: UTC (YYYYMMDDTHHMMSS)

Correlations: 6 highest correlation values (0 to 255) which translates to a percentage (0 to 100%) correlation data are received from the sensor in hexadecimal format but converted to an 8 bit unsigned integer for science log decoding.

Sample: 6 associated correlations sample index within the listening window (up to 65535). Sample index data are received from the sensor in hexadecimal format but converted to a 16 bit unsigned integer for science log decoding.

An example row of RAFOS sensor data extracted from a science_log.csv file is shown in Figure A-13.

RAFOS	20190712T130756	34	315	31	281	31	256	30	61	29	103	29	410

Figure A-13: Example Row of RAFOS Sensor Data in a science_log.csv File

For more information on interpreting the science_log.bin file along with additional examples, refer to "Converting the science_log.bin and vitals_log.bin Files to CSV Files" on page 4-19.

A.10.4 Show Command

The **Show** command displays all of the information available from the Seascan RAFOS sensor. To use the **Show** command, a PC must be connected to the float and running a terminal program.

To display the RAFOS sensor information, enter:

Syntax: `rafos_show`

A.10.5 Other Commands

The following additional Seascan RAFOS sensor commands are available:

rafos_config. Configures all the Seascan RAFOS sensor configuration parameters. The RAFOS sensor is preconfigured and has no settings to change.

Syntax: `rafos_config`

Rafos_listen. Starts a RAFOS sensor listening window for the specified amount of time and collects the results. (default is 3 minutes)

Syntax: `rafos_listen [duration_in_seconds]`

rafos_test. Runs through configuration, displays configuration, listens for correlations for 3 minutes, and obtains correlation data from the RAFOS sensor. This test is intended to be run with a sound source and therefore the test will fail unless at least one correlation of above 75% is detected. The sys_self_test version of the test will skip the correlation check since the sound source is not expected to be on during the full system self test.

Syntax: rafos_test

For example,

```
>rafos_test
20190712T130448|5|T_CMD|rafos_test
20190712T130448|5|RAFOS|Seascan RAFOS Test:
20190712T130448|5|RAFOS|configure|preconfigured
20190712T130448|5|RAFOS| Seascan RAFOS Configure : PASS
20190712T130448|5|RAFOS|sensor has no settings to obtain
20190712T130448|5|RAFOS| Seascan RAFOS Obtain Settings : PASS
20190712T130448|5|RAFOS|sensor has no settings to display
20190712T130451|5|RAFOS|listening for 180 seconds
RAFOS|data=1|Time:1562936875|C1:27|S1:006A|C2:27|S2:0153|C3:26|S3:01FD
RAFOS|data=2|C4:24|S4:01EE|C5:23|S5:0241|C6:21|S6:01CB
20190712T130756|5|RAFOS|correlation1: 27 (15.3%), sample1: 006A (106)
20190712T130756|5|RAFOS|correlation2: 27 (15.3%), sample2: 0153 (339)
20190712T130756|5|RAFOS|correlation3: 26 (14.9%), sample3: 01FD (509)
20190712T130756|5|RAFOS|correlation4: 24 (14.1%), sample4: 01EE (494)
20190712T130756|5|RAFOS|correlation5: 23 (13.7%), sample5: 0241 (577)
20190712T130756|5|RAFOS|correlation6: 21 (12.9%), sample6: 01CB (459)
20190712T130756|5|RAFOS| Seascan RAFOS Obtain Correlation : PASS
```

```
20190712T130756|3|RAFOS|detected 0 correlations
20190712T130756|3|RAFOS| Seascan RAFOS Check Correlation : FAIL
20190712T130756|3|RAFOS| Seascan RAFOS Test : <<FAIL>>
Seascan RAFOS test FAILED
```

A.10.6 RAFOS Sensor Factory Configuration

The factory configuration of the installed Seascan RAFOS sensor is as provided by the manufacturer.

A.11 Seascan RTC

The Teledyne Webb Research APEX profiling float optionally includes the Seascan RTC low power high accuracy real time clock module. The RTC is designed to be a stop watch for critical timing events onboard the float such as RAFOS listening windows. The RTC is internally mounted and connected.

A.11.1 Specifying the RTC Hardware Configuration

A line in the sensors.cfg file defines the RTC hardware configuration:

```
sensor seascan_rtc COM5 B9600
```

For more information on the sensors.cfg file along with additional examples, refer to “sensors.cfg File” on page 4-16.

A.11.2 Specifying the RTC Sampling Behavior

The interval sampling behavior for the RTC is specified using a single entry in the sample.cfg file for each of any number of pressure ranges in accordance with the following syntax:

SAMPLE CLK *[<start> [<stop> [<interval> [<units> [<count>]]]]]*

For example, to sample the CLK every 10 dbar beginning at 1000 dbar and ending at 200 dbar, enter:

```
<ASCENT>
SAMPLE CLK 1000 200 10 DBAR
```

NOTE *The main application for the Seascan RTC is in conjunction with the Seascan RAFOS sensor which will automatically record Seascan RTC times tamps as needed and no additional sampling is required.*

For more information on using the sample.cfg file to program the behavior of the float along with additional examples, refer to “sample.cfg File” on page 4-2.

A.11.3 Reading the Science Log RTC Data

RTC data are provided in the science_log.bin file. For instructions on how to uncompress this file and convert it to a CSV file, refer to “Converting the science_log.bin and vitals_log.bin Files to CSV Files” on page 4-19.

All rows in the science_log.csv file with the first column label "SRTC" contain RTC data. The columns are from left to right:

Sensor type:	SRTC
Time stamp:	UTC (YYYYMMDDTHHMMSS)
Seacan RTC Time:	Time in seconds since the last time the clock was reset.

An example row of RTC data extracted from a science_log.csv file is shown in Figure A-14.

SRTC	20190702T213324	98337

Figure A-14: Example Row of RTC Data in a science_log.csv File

For more information on interpreting the science_log.bin file along with additional examples, refer to “Converting the science_log.bin and vitals_log.bin Files to CSV Files” on page 4-19.

A.11.4 Show Command

The **Show** command displays all of the information available from the Seascan RTC. The RTC is preconfigured but this command will report the crystal calibration information. To use the **Show** command, a PC must be connected to the float and running a terminal program.

To display the RTC information, enter:

Syntax: `srtc_show`

A.11.5 Other Commands

The following additional Seascan RTC commands are available:

srtc_config. Configures all Seascan RTC configuration parameters. The RTC is preconfigured and has no settings to change.

Syntax: `srtc_config`

srtc_test. Runs through configuration, displays configuration and obtains the current RTC timestamp.

Syntax: `srtc_test`

For example,

```
> srtc_test
20190712T130844|5|T_CMD|srtc_test
20190712T130844|5|test|Seascan RTC Test:
20190712T130844|5|SRTC|configure|preconfigured
```

```
20190712T130844|5|test| Seascan RTC Configure : PASS
20190712T130847|5|test| Seascan RTC Obtain Settings : PASS
20190712T130847|5|SRTC|Crystal Corrections: 00000000 01 1C60 FFFDAB97
SRTC|Timestamp:1562936935|RTC_TIME:1420
20190712T130855|5|test|Seascan RTC time: 1420
20190712T130855|5|test| Seascan RTC Obtain Sample : PASS
20190712T130855|5|test| Seascan RTC Test : <<PASS>>
Seascan RTC test PASSED
```

srtc_reset. Resets the Seascan RTC to 0. It is recommended to run this command with a verbosity of 7 and a terminal program to time stamp lines received for an accurate time of when the clock was reset.

Syntax: `srtc_reset`

For example,

```
> srtc_reset
20190802T125711|5|T_CMD|srtc_reset
20190802T125720|5|SRTC|reset_time|passed
Seascan RTC reset PASSED
```

A.11.6 RTC Factory Configuration

The factory configuration of the installed Seascan RTC is as provided by the manufacturer.

APPENDIX B: Ice Avoidance

For Teledyne Webb Research APEX profiling floats that are expected to be deployed in or to drift into regions near the Arctic or Antarctic where surface ice can be present, an ice avoidance function can be enabled before or after deployment. Activation of ice avoidance enables the float to detect the presence of ice and to respond appropriately, preventing catastrophic damage by either being trapped in the ice or by being crushed by floating ice during ice breakup. If ice is determined to be present, ice avoidance will modify the behavior of the float during both the Ascent and Surface phases of each profiling cycle in accordance with the settings of specific configuration parameters in the mission.cfg and sample.cfg files.

When, and only when ice avoidance is enabled for a particular month, the float will check for the following four conditions during each profiling cycle:

- Surface ice layer
- Ice breakup period
- Surface ice cap
- Ice descent and ice ascent cycles

For each of these conditions, the float will avoid the surface and record the event to the system_log.txt file. All data collected during the current profiling cycle will be recorded to the science_log.bin and vitals_log.bin files for transmission during the next Surface phase for which communications are successful.

Shown in Figure B-1 is an example of float behavior when ice avoidance is enabled and surface ice is detected followed by an ice breakup period and detection of a surface ice cap. In a case where ice is present but *is not* being detected by the float and telemetry fails, and *only* for the months when ice avoidance is enabled, the float can be configured to perform a number of Ice Descent and Ice Ascent cycles between telemetry retries.

B.1 Detecting a Surface Ice Layer

The float will collect temperature data beginning at a detection pressure of typically 50 dbar at specified intervals of typically 2 dbar while ascending through the mixed layer under surface ice. After reaching an evasion pressure of typically 20 dbar, the median temperature is determined and compared to a threshold temperature below which ice is expected to be present. If the presence of ice is detected, the float will stop its ascent and transition to the Park Descent phase of a new profiling cycle. If the presence of ice is not detected, the float will continue to the surface.

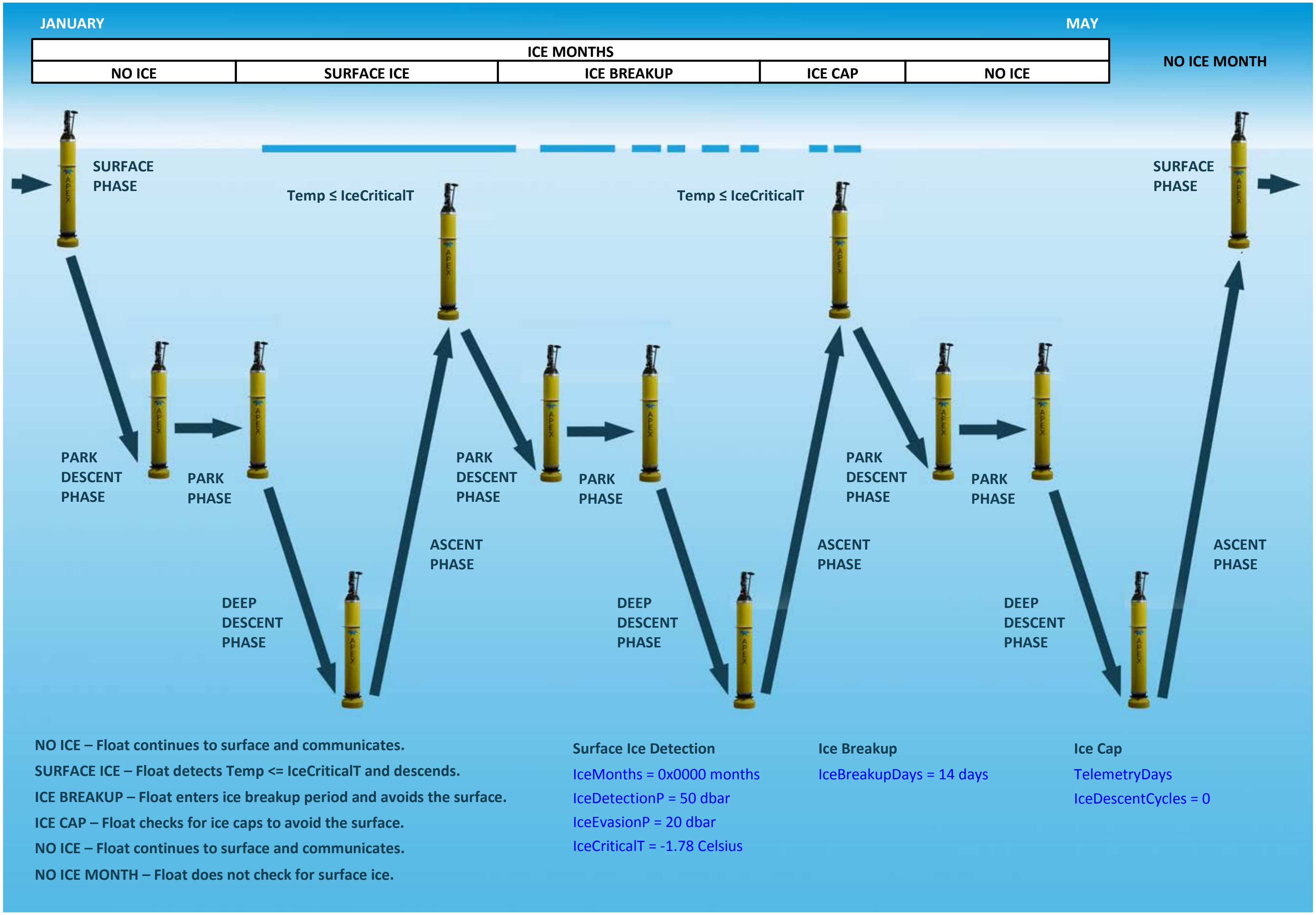


Figure B-1: Example Ice Avoidance Behavior of Float when Surface Ice is Detected followed by an Ice Breakup Period and an Ice Cap

The mission.cfg file configuration parameters that apply to detecting surface ice are **IceMonths**, **IceDetectionP**, **IceEvasionP**, and **IceCriticalT**:

IceMonths. Determines the months of the year during which ice avoidance is enabled. To determine the numerical entry, which is in hexadecimal, select one or more months during which to enable ice avoidance, where each month is a "one" bit in a 12-bit binary string representing January (000000000001) to December (100000000000). For more than one month, whether contiguous or not, "or" the values of all the months to determine the binary value. For example, to detect ice during the contiguous months of May to September, the binary value is 000111110000. Converting to hexadecimal, the entry is 0x01f0. For the months January to May and September to December, the binary value is 111100011111. Converting to hexadecimal, the entry is 0x0f1f. When set to “0000,” which is the default, all of the configuration parameters pertaining to the ice avoidance function become hidden and therefore are not displayed in the log file output of the mission and sample configuration parameters as the ice avoidance function is disabled.

Syntax:	IceMonths *<months>*
Range:	0x0000 (never enabled)– 0x0fff (always enabled)
Default:	0x0000 months

CAUTION *Unless the intent is for the float to remain under the ice all year long, ice avoidance should be disabled for at least one month a year to allow the float to attempt to reach the surface and transmit its data and position and to receive new mission.cfg and sample.cfg files if available. If ice avoidance is not disabled for at least one month, the float may never reach the surface. This period should be when there is the least likelihood of the presence of ice.*

NOTE ***IceMonths*** *applies to the start of the ascent. Once the float has begun its ascent, if the month changes, the float will continue to use the* ***IceMonth*** *configuration set at the start of the ascent.*

IceDetectionP. The pressure in decibars at which temperature data collection begins during the Ascent phase for the purpose of determining the mixed layer median temperature. **IceDetectionP** must be at least 15 dbar greater than **IceEvasionP**.

Syntax:	IceDetectionP *<decibars>*
Range:	15–100 dbar
Default:	50 dbar

IceEvasionP. The pressure in decibars at which the float begins processing the collected temperature data to determine the mixed layer median temperature. While processing the temperature data, the float continues to collect more data as it ascends. If the median temperature is at or below that of **IceCriticalT**, the ascent is halted and the float transitions to the Park Descent phase of a new profiling cycle. **IceEvasionP** must be at least 15 dbar less than **IceDetectionP**.

Syntax: IceEvasionP *<decibars>*
Range: 0–85 dbar
Default: 20 dbar

IceCriticalT. The water temperature in Celsius below which ice is determined to be present.

Syntax: IceCriticalT *<Celsius>*
Range: -3–30 Celsius
Default: -1.78 Celsius

NOTE *Thermal detection is typically performed based on the sample.cfg configuration for temperature sampling. However, if the float detects that it has not collected a thermal sample in over 60 seconds, it will solicit a CTD temperature sample for thermal detection.*

NOTE *The float will continue to collect temperature samples as it ascends above **IceEvasionP**. However, the float can be configured with a command to check the temperature once at **IceEvasionP** only.*

B.2 Commencing the Ice Breakup Period

Following previous periods of ice detection, if no surface ice is detected in the current profiling cycle, the float will commence the ice breakup period. During this time, the float will continue to avoid the surface and transition to the Park Descent phase for each new profiling cycle. After the ice breakup period ends, the float will attempt to get to the surface unless it determines the presence of surface ice.

The only mission.cfg file configuration parameter that applies to detecting ice breakup is **IceBreakupDays**:

IceBreakupDays. The period in days over which the float avoids the surface due to the possible presence of large, crushing icebergs but performs all the other phases during each profiling cycle. The period starts with the first determination of the *non*-presence of ice after having determined the presence of ice over the previous profiling cycles. The setting is typically a multiple of the profiling cycle duration.

Syntax: IceBreakupDays *<days>*
Range: 0–30 days
Default: 14 days

IceBreakupDays may coincide with the length of the profiling cycle. If the profiling cycle is 10 days, an **IceBreakupDays** of 10 would indicate determining if ice was present on the previous profiling cycle. However, depending on the desired behavior of the float, **IceBreakupDays** could be set to one additional day or one less day to cover the condition if the profiling cycle is slightly more or less.

B.3 Detecting a Surface Ice Cap

Following the end of the ice breakup period, the float will attempt to surface and communicate over the Iridium satellite network, depending on **TelemetryDays** and **IceDescentCycles**. For information on **TelemetryDays**, refer to Surface Phase on page 3-16. For information on **IceDescentCycles**, refer to “Ice Descent and Ice Ascent Cycles” below.

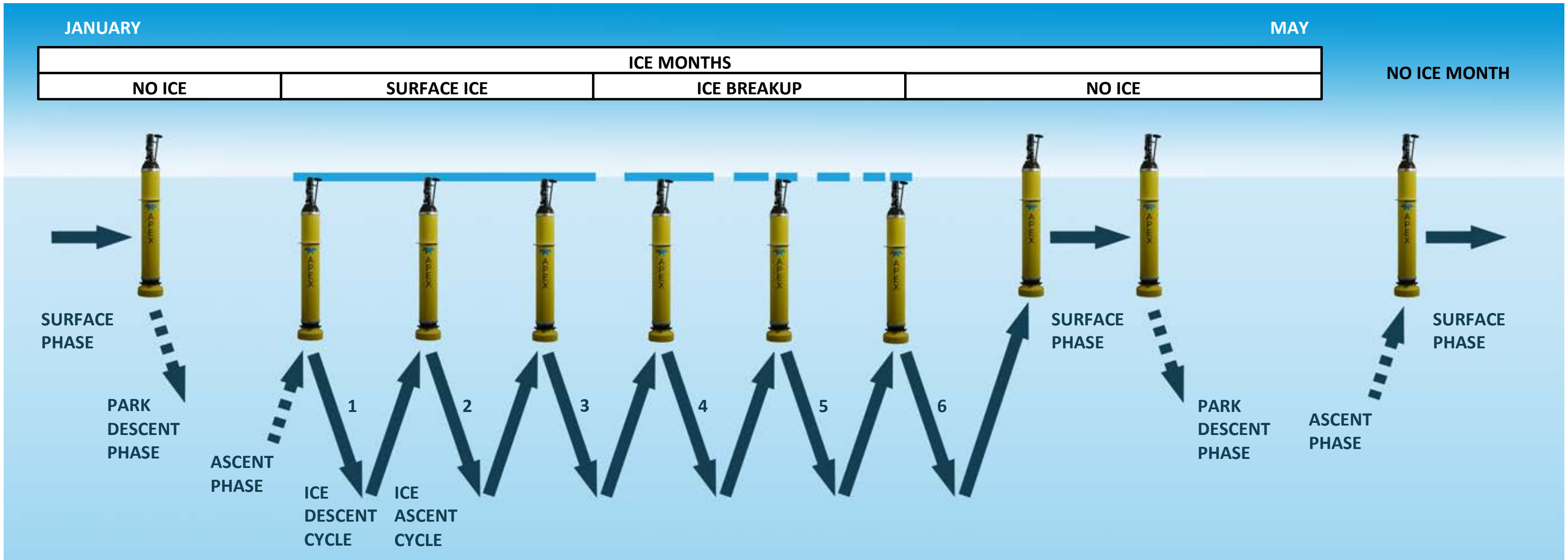
When **IceDescentCycles** is disabled, it is possible that the float may encounter a surface ice cap that prevents it from reaching the surface to communicate. During the Surface phase, the float will attempt to communicate until the end of **AscentTimeout**. After **AscentTimeout**, should communications continue to fail, and the mixed layer median temperature is at or below that of **IceCriticalT**, there may be a surface ice cap present. Should this be the case, the float will abort the Surface phase before the **UpTime** expiration and transition to the Park Descent phase of a new profiling cycle. For information on **AscentTimeout** and **UpTime**, refer to “Ascent Phase” on page 3-14.

If **IceDescentCycles** is *disabled*, the float *will* try to detect an ice cap if it cannot find the ski. If **IceDescentCycles** is *enabled*, the float *will not* try to detect the ice cap and will operate in accordance with “Ice Descent and Ice Ascent Cycles” described below where it tries to drift out from under an ice cap.

If **TelemetryDays** is *disabled*, the float will try to detect an ice cap if it cannot perform telemetry. If **TelemetryDays** is *configured*, the float will try to detect an ice cap for the days within the telemetry season, and for the days outside of the telemetry season, the float will automatically descend without trying to detect an ice cap.

B.4 Ice Descent and Ice Ascent Cycles

If there is no ice, the float will be able to ascend to the surface and perform telemetry. However, there could be a case where there is ice and the float is not detecting it, making it unable to perform telemetry. Should this be the case, and *only* for the months when ice avoidance is enabled, the float can be configured to perform multiple Ice Descent and Ice Ascent cycles during which at some point it might drift past the ice and ascend all the way to the surface on the next Ice Ascent cycle as shown in Figure B-2. In an Ice Descent cycle the float descends by a specified number of decibars for a specified duration, after which it starts the Ice Ascent cycle where it ascends in an attempt to reach the surface and retry telemetry.



Ice Descent and Ice Ascent Cycles

IceDescentCycles = 8
IceDescentPressure = 100 dbar
IceDescentCount = 2000
IceDescentTimerInterval = 300 sec
IceDescentNudge = 16 counts
IceDescentTimeout = 42 min
IceAscentTimeout = 62 min

Ice Ascent Phase

InitialBuoyancyNudge = 300 counts
AscentRate = 0.08 dbar/sec
BouyancyNudge = 50 counts
AscentTimerInterval = 120 sec

NO ICE – Float continues to surface and communicates.
SURFACE ICE – Float commences Ice Descent and Ice Ascent cycles.
ICE BREAKUP – Float continues Ice Descent and Ice Ascent cycles.
NO ICE – Float continues to surface and communicates.
NO ICE MONTH – Float does not check for surface ice.

Figure B-2: Example Ice Descent and Ice Ascent Cycles

The mission.cfg file configuration parameters that apply to the Ice Descent cycle are **IceDescentCycles**, **IceDescentPressure**, **IceDescentCount**, **IceDescentTimerInterval**, **IceDescentNudge**, **IceDescentTimeout**, and I**ceAscentTimeout**:

IceDescentCycles. The *maximum* number of Ice Descent and Ice Ascent cycles the float will perform when it is unable to perform telemetry. When it has completed the specified number of cycles, the float will continue with the next profiling cycle.

Syntax:	IceDescentCycles <*n*>
Range:	0–24 cycles
Default:	0 (disabled)

IceDescentPressure. The pressure associated with **IceDescentCycles**. The float descends from its current depth by the number of decibars specified, limited by the maximum descent depth which is 2000 dbar. For example, if **IceDescentPressure** is set to 100 dbar and the float starts descending from 1950 dbar, it would not descend to 2050 dbar. It would target 2000 dbar to avoid going below the maximum pressure.

Syntax:	IceDescentPressure <*decibars*>
Range:	0–300 dbar
Default:	100 dbar

IceDescentCount. The delta buoyancy position in counts that will cause the float to descend by **IceDescentPressure**. The float moves the current buoyancy position by the amount specified. This configuration parameter is dynamically updated during each Ice Descent cycle in accordance with **IceDescentNudge**.

Syntax:	IceDescentCount <*counts*>
Range:	**buoyancy_pump_min** and **buoyancy_pump_max** counts
Default:	2000 counts

IceDescentTimerInterval. The maximum interval in seconds between float samples of pressure to determine if the float has descended by the number of decibars specified by **IceDescentPressure**. The float may wake up at a different interval based on the current descent rate.

Syntax:	IceDescentTimerInterval <*seconds*>
Range:	5–5400 sec (90 minutes)
Default:	300 sec

IceDescentNudge. The number of counts to remove from the current buoyancy position so as to incrementally decrease the buoyancy of the float to descend by **IceDescentPressure**.

Syntax:	IceDescentNudge *<counts>*
Range:	0–2000 counts
Default:	16 counts

IceDescentTimeout. The amount of time in minutes allowed for the float to descend from its current depth by the number of decibars specified by **IceDescentPressure**. Should the float not reach this depth in this period, it will begin the Ice Ascent cycle. **IceDescentTimeout** is also used to manage the float's descent rate. The float will attempt to descend by **IceDescentPressure** just before **IceDescentTimeout**, which should be set to prevent the float from having to pull in an excess amount of oil to descend by **IceDescentPressure** too quickly and have to pump that extra oil back out, but not too long where the descent would not be completed. A default value of 42 minutes was chosen based on the default **IceDescentPressure** of 100 dbar assuming it would take approximately 40 minutes to descend and approximately 20 minutes to ascend.

Syntax:	IceDescentTimeout *<minutes>*
Range:	1–600 min (10 hours)
Default:	42 min

IceAscentTimeout. The amount of time in minutes allowed for the float to ascend from its current depth and reach the surface. Should the float not reach the surface in this period, it will begin the Surface Phase and retry telemetry. It is recommended that **IceAscentTimeout** be set to greater than **IceDescentTimeout** to allow the float to complete the ascent to the surface if there is no longer any ice.

Syntax:	IceAscentTimeout *<minutes>*
Range:	1 – 600 min (10 hours)
Default:	62 min

Additional mission.cfg file configuration parameters that apply to the Ice Ascent cycle are **InitialBuoyancyNudge**, **AscentRate**, **BuoyancyNudge** , and **AscentTimerInterval**. For information on these configuration parameters, refer to "Ascent Phase" on page 3-14.

<u>Example 1</u>. Refer to Figure B-3. During a month when ice avoidance is enabled as determined by **IceMonths**, the float ascends from the Profile depth and encounters ice at 250 meters. At the end of **AscentTimeout** the float transitions to the Surface phase and attempts telemetry. Because the float will not be able to transmit the data files, it will begin the Ice Descent/Ice Ascent cycles as determined by **IceDescentCycles**, which has been set to 6. The float descends by **IceDescentPressure** and then ascends to retry telemetry. In this example, on the third Ice Descent/Ice Ascent cycle, the float reaches the surface before the end of **IceAscentTimeout**, transmits the data files and begins the next profiling cycle.

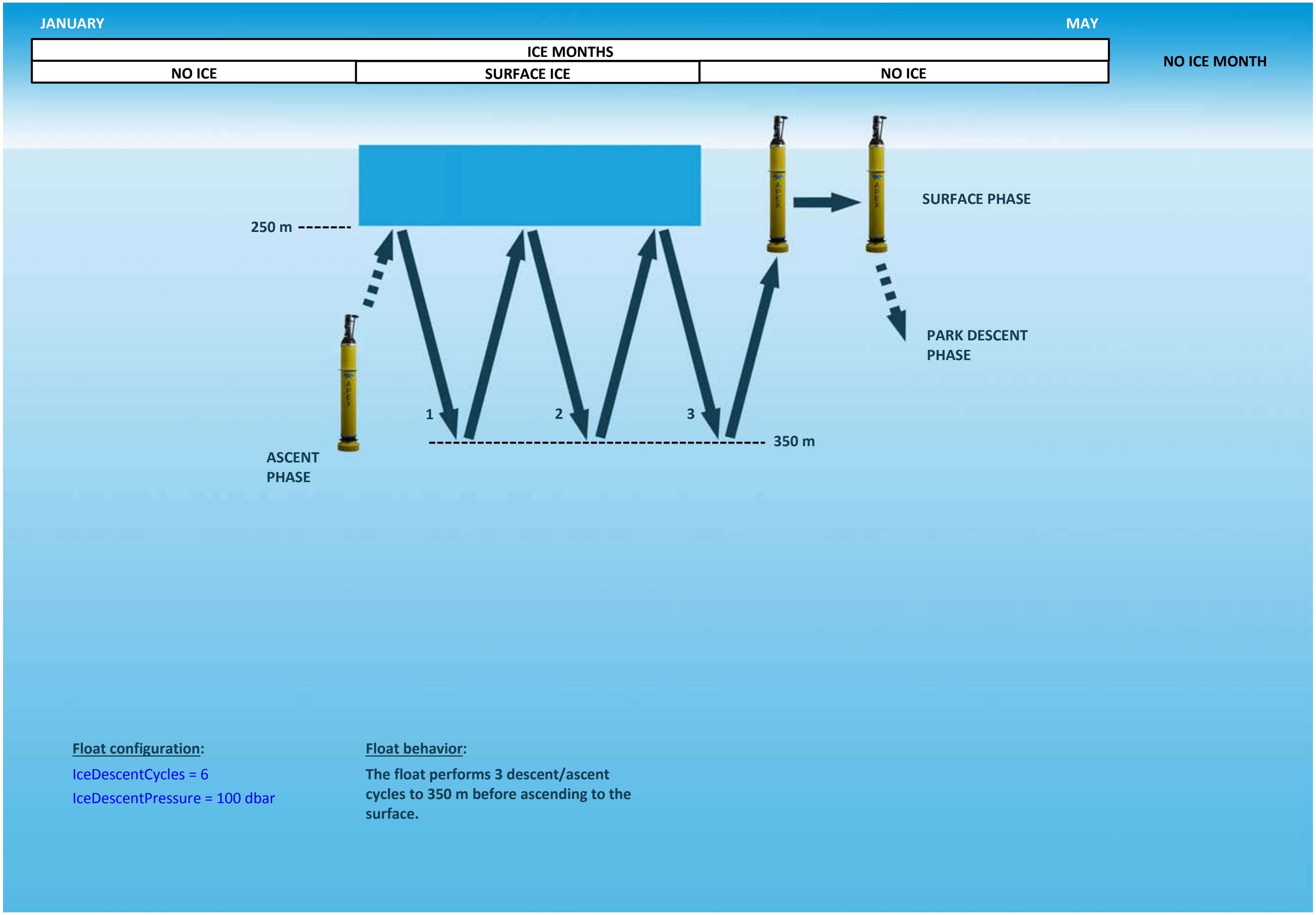


Figure B-3: Ice Descent and Ice Ascent Example 1

Example 2. Refer to Figure B-4. During a month when ice avoidance is enabled as determined by **IceMonths**, the float ascends from the Profile depth and encounters ice at 250 meters. At the end of **AscentTimeout** the float transitions to the Surface phase and attempts telemetry. Because the float will not be able to transmit the data files, it will begin the Ice Descent/Ice Ascent cycles as determined by **IceDescentCycles**, which has been set to 6. The float descends by **IceDescentPressure** and then ascends to retry telemetry. In this example, on the third Ice Descent/Ice Ascent cycle, the float encounters but *does not* detect another ice mass, this time near the surface. The float continues with the next 2 Ice Descent/Ice Ascent cycles where on the fifth Ice Ascent cycle, the float reaches the surface before the end of **IceAscentTimeout**, transmits the data files and begins the next profiling cycle.

B.5 Sampling

The float is capable of sampling during the Ice Descent and Ice Ascent cycles as specified in the sample.cfg file. When the float travels through **IceDetectionP** and **IceEvasionP** it will continue to use temperature samples for ice detection. If ice is detected the float will abandon the Ice Descent cycles and continue to the next profiling cycle.

Syntax: *<phase>*
SAMPLE *<sensor type> [<start> [<stop> [<interval> [<units> [<count>]]]]]*

phase: ICEDESCENT
ICEASCENT

sensor type: CTD - conductivity, temperature and pressure sensor
LGR - Multichannel CTD
OPT - oxygen sensor
FLBB - fluorometer

start: Start pressure (0–2500) dbar

stop: Stop pressure (0–2500) dbar

interval: Pressure interval (0-100) dbar

units: DBAR or SEC

count: Number of samples

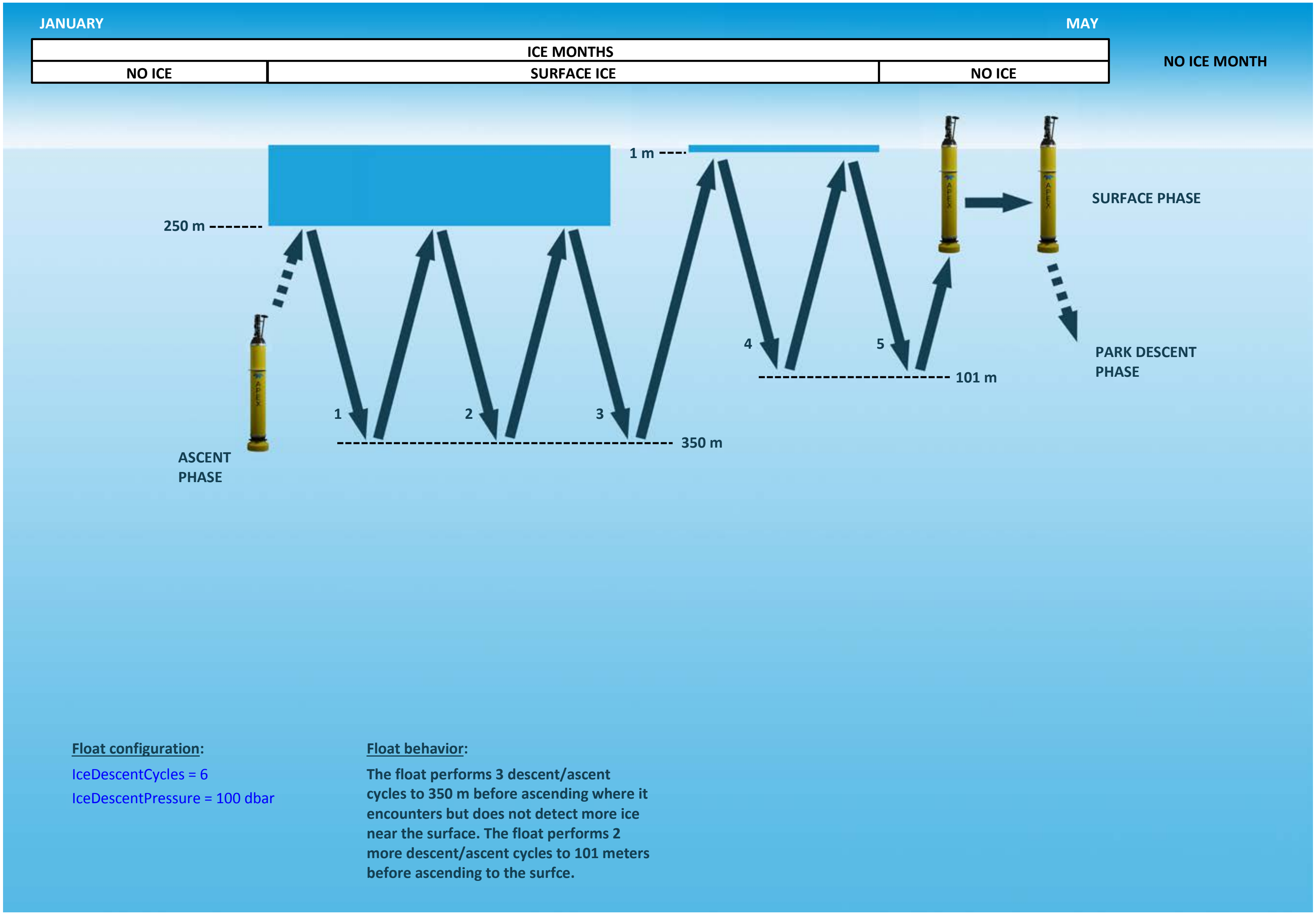


Figure B-4: Ice Descent and Ice Ascent Example 2

APPENDIX C: Time of Day Operation

Time of day operation enables the sampling for each profiling cycle to coincide with specific depths and times of the day. Time of day operation is enabled with a setting for the **AscentStartTimes** configuration parameter. For information on the **AscentStartTimes** configuration parameter, refer to “Ascent Phase” on page 3-14.

C.1 *Standard* Time of Day Operation

Standard time of day operation specifies the time of day at which the float will start its ascent after completing the Park and Deep Descent phases. The time of day is specified by the operator with **AscentStartTimes** which is in minutes after Greenwich Mean Time (GMT) midnight. The float remains at the Park depth for a period based on **DownTime**, the actual Park decent time and the amount of time specified by **DeepDescentTimeout** and **AscentStartTimes**. The settings of these configuration parameters account for the time the float needs to travel to the Profile depth to be ready to start the ascent at the specified time of day. For example, the float could be scheduled to start sampling at 6:00 a.m. at 200 dbar. However, since the time of day specifies the start of the ascent, the float’s ascent rate must be taken into account when determining **AscentStartTimes**.

C.2 *Extended* Time of Day Operation

Time of day operation allows for *additional* **AscentStartTimes** configuration parameter settings to be specified for the same day. The float will automatically adjust the period of each of the corresponding Park phases based on the actual time of day that it reached the Park depth, the time of day at which it must be at the Profile depth to start the next Ascent phase, and **DeepDescentTimeout**. For these additional Ascent phases, **DownTime** does not apply. For example, the float could be configured to ascend twice in the same day, such as sunrise and sunset, after parking for several days.

C.3 Unreachable Time of Day Operation

If there is not enough time for the float to descend, to park and to do a deep descent such as to meet the time of day for the ascent, the float will park for an extended amount of time to meet the time of day requirement for the following day.

C.4 Configuration Parameters that Affect Time of Day Operation

The following configuration parameters affect the time of day operation, and their settings should be modified as might be required when enabling time of day operation:

DeepProfileFirst. When set to “on,” the float will not use time of day operation during the first profiling cycle. Sampling will be in accordance with the sample.cfg file at the specified depths without regard to the time of day. Time of day operation will be used on subsequent profiling cycles.

DownTime. Only used for profiling cycles that use the <minutes1> setting of the **AscentStartTimes** configuration parameter.

AscentRate. Needs to be taken into account by the operator when determining the settings for **AscentStartTimes**. **AscentRate** determines how long it will take the float to ascend from the Park or the Profile depth to the surface or to the sampling depths in the sample.cfg file. It is set by the operator, and the default setting is 0.08 dbar/sec.

DeepDescentTimeout. Should be set to the expected time for the float to descend from the Park depth to the Profile depth based on **ParkPressure** and **DeepDescentPressure**. Setting **DeepDescentTimeout** in this manner minimizes or removes any time variability in the Deep Descent phase. Variability of the time in the Park Descent and Surfaces phases is automatically compensated for in the Park phase, but some time variability will remain in the Ascent phase, especially if the float encounters a mixed layer near the surface, as a mixed layer may affect the ascent rate of the float.

CAUTION *If **DeepDescentTimeout** is much less or much greater than the actual descent time, unexpected time of day operation behavior could occur. If set to be much less, the float will start ascending from a shallower depth, the ascent time will be less, and the float may arrive earlier than expected. If set to be much greater, the float will start its descent from the Park depth earlier than expected, reaching the Profile depth earlier than expected and starting its ascent earlier than expected.*

PnPCycleLen. Enables the float to ascend from the Park depth one or more times before ascending from the Profile depth. With **PnPCycleLen** greater than 1, and when time of day is enabled, sampling for all the ascents can be made to coincide with specific depths and times of the day on the same day for each profiling cycle. However, the time for the float to ascend from the Profile Depth is always longer than the time to ascend from the Park depth. This longer time must be taken into account when determining **PnPCycleLen** and **AscentStartTimes**. With **PnPCycleLen** set to 1, which is both the default and typical setting, the float descends to the Profile depth from the Park depth for every profiling cycle. For more information on **PNPCycleLen**, refer to "Deep Descent Phase" on page 3-9.

C.5 Example Time of Day Operation

As an example, to illustrate standard and extended time of day operations, an operator might want to profile from 50 dbar during the Ascent phase at four specific times a day. Sampling could begin, for example, at 6:00 a.m. (GMT 06:00), 12:00 noon (GMT 12:00), 6:00 p.m. (GMT 18:00), and 12:00 midnight (GMT 24:00) of the same day. For this example, the float will park at 500 dbar and descend to 600 dbar before starting the Ascent phase. In between the multiple profiling sessions, the float will remain in the Park phase for 5 days and sample on the 6th day. This scenario is illustrated in Figure C-1.

From a 600-dbar Profile depth, ascending at a rate of 0.08 dbar per second to 50 dbar would take the float (600 – 50)/0.08 = 115 minutes.

- If the float needs to be at 50 dbar at GMT 6:00 to sample, **AscentStartTime** would be set to 06:00 – 115 minutes or GMT 4:05 or 245 minutes after GMT 00:00.
- If the float needs to be at 50 dbar at GMT 12:00 to sample, **AscentStartTime** would be set 12:00 – 115 minutes or GMT 10:05 or 605 minutes after GMT 00:00.
- If the float needs to be at 50 dbar at GMT 18:00 to sample, **AscentStartTime** would be set to 18:00 – 115 minutes or GMT 16:05 or 965 minutes after GMT 00:00.
- If the float needs to be at 50 dbar at GMT 24:00 a.m. to sample, **AscentStartTime** would be set to 24:00 – 115 minutes or GMT 22:05 or 1325 minutes after GMT 00:00.

The applicable configuration parameter settings would be as follows:

AscentStartTimes: 245 605 965 1325
Downtime: 7200 minutes (5 days)
ParkPressure: 500 dbar
DeepDescentPressure: 600 dbar
PnPCycleLen: 1
DeepProfileFirst: off
AscentRate: 0.08 dbar/sec

The estimated float travel and park time calculations for the first ascent at 245 minutes after GMT 00:00 are shown in Table C-1.

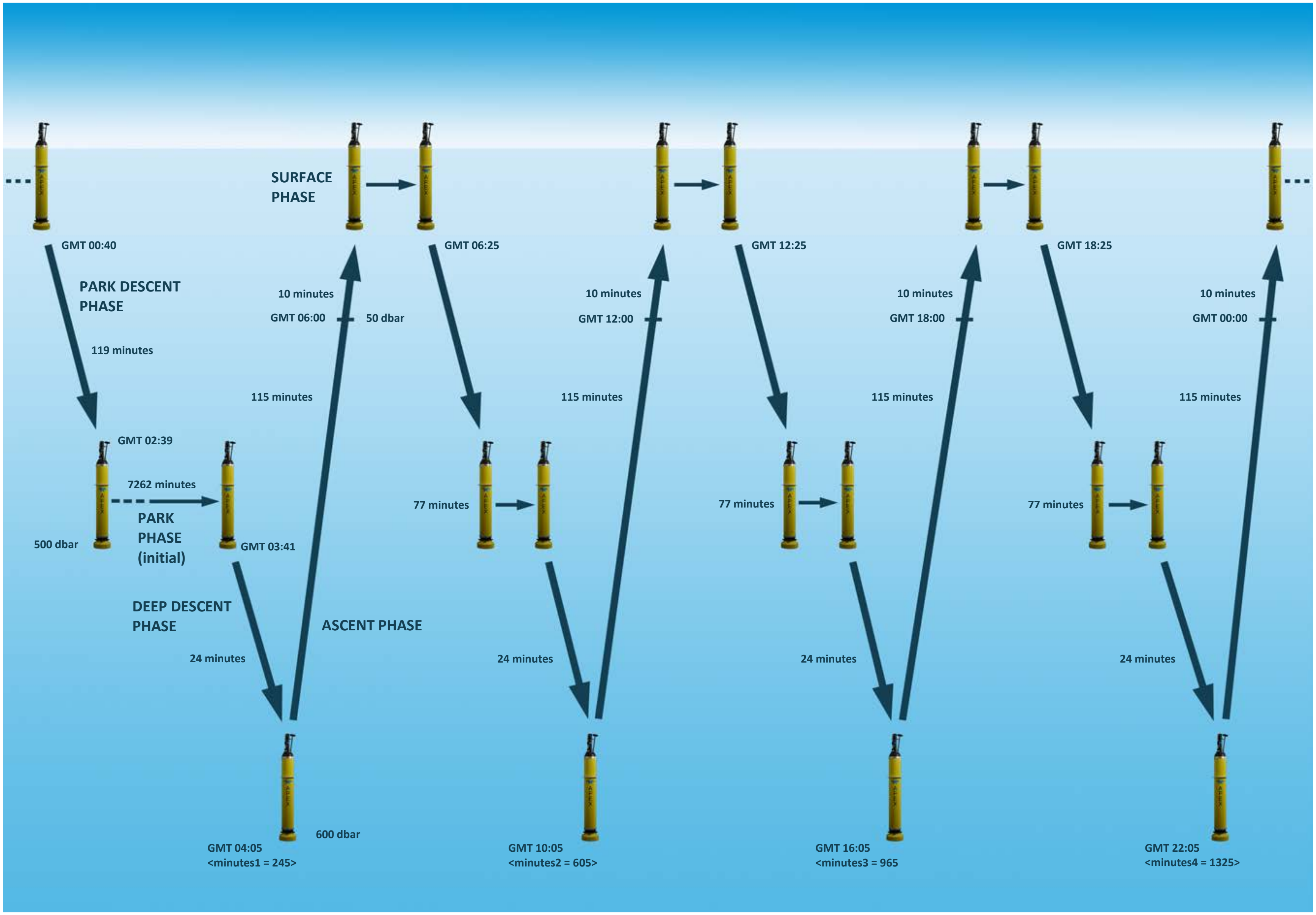


Figure C-1: Example Time of Day Operation with Four Profiling Cycles

Table C-1: Estimated Float Travel and Park Time Calculations for the First Ascent at 245 Minutes after GMT 00:00

Mission Leg	Formula	Result
Park descent travel time	500 dbar/0.07 dbar per second	119 minutes
Park time length	7200 – 119 – 24 + 205 minutes	7262 minute (See Note below)
Deep descent travel time	100 dbar/0.07 dbar per second	24 minutes
Ascent leg travel time to 50 dbar	550 dbar/0.08 dbar per second	115 minutes
Ascent leg travel time from 50 dbar to surface	50 dbar/0.08 dbar per second	10 minutes

NOTE *Without time of day operation, the float would have ended the Park phase and started descending to the Profile depth at GMT 00:16, traveling for 24 minutes to reach it at GMT 00:40. However, in order to reach 50 dbar at GMT 06:00, the float needs to wait in park for an additional 205 minutes to begin the descent to the Profile depth at GMT 03:41. Therefore the float will add 205 minutes to the Park time based on* ***AscentStartTimes*** *such that the ascent will begin at GMT 4:05. This additional Park time is automatically calculated and applied by the float for each profiling cycle, as the time the float leaves the surface can vary for each profiling cycle.*

In in Table C-1 the Mission Leg parameters are as follows:

Park descent travel time: The elapsed time for the float to travel to the 500 dbar park depth at an approximate rate of 0.07 dbar per second.

Park time length: The length of time the float remains in the Park phase as determined by **DownTime**, the actual park decent time, **DeepDescentTimeout**, and the time of day adjustment based on **AscentStartTimes**.

Deep descent travel time: The elapsed time for the float to travel from the 500-dbar Park depth to the 600-dbar Profile depth at a rate of 0.07 dbar per second.

Ascent leg time travel time to 50 dbar: The elapsed time for the float to travel from the 600-dbar Profile depth to the desired 50-dbar time of day depth at an ascent rate of 0.08 dbar per second.

Ascent leg time travel from 50 dbar to surface: The elapsed time for the float to travel from the 50-dbar depth to the surface at an ascent rate of 0.08 dbar per second.

The estimated float mission GMT times based on the float leaving the surface at 00:40 GMT are shown in Table C-2. The actual time that the float leaves the surface is variable.

Table C-2: Estimated Float Mission GMT Times Based on Float Leaving the Surface at 00:40 GMT

Mission Leg	Formula	Mission GMT Times
Park descent start time	The time the float starts the Park Descent phase	GMT 00:40
Desired sample start time	The desired time to start sampling at 50 dbar at GMT 6:00	GMT 06:00
Ascent start time	06:00 – 115 minutes = 04:05	GMT 04:05
Deep descent start time	04:05 – 24 minutes = 03:41	GMT 03:41
Park phase start time	00:40 + 119 minutes = 02:39	GMT 02:39
Normal Park phase end time	2:39 + 4 days 21 hours 37 minutes = 00:16	GMT 00:16
Park phase adjusted for time or day	03:41 – 0:16 = 3 hours and 25 minutes	—

For each of the remaining three ascents in this example, the Park phase period is set by the float based on the time of day it is to reach the Profile depth at the <minutes2>, <minutes3> and <minutes4> settings of **AscentStartTimes**, the actual time of day it begins each Park Descent phase, the 119 minutes required to reach the Park depth, and the 24 minutes required to reach the Profile depth. For example, it will take the float 10 minutes to reach the surface from 50 dbar, placing the float at the surface at GMT 06:10 after the first ascent. Should it take 15 minutes for telemetry, the float will begin the next Park Descent phase at GMT 06:25. It must be at the Profile depth 220 minutes later at GMT 10:05 to begin sampling on the ascent at GMT 12:00, as the <minutes2> setting of **AscentStartTimes** is 605. With 119 minutes required to reach the Park depth from the surface and 24 minutes to reach the Profile depth from the Park depth, the Park phase period must be 77 minutes:

$$220 - 119 - 24 = 77$$

For the remaining two ascents, the Park phase period will be adjusted in a similar manner and can vary depending on the time of day at which the float actually leaves the surface. At the completion of the Surface phase after the last ascent, the profiling cycles repeat with the initial Park phase of approximately 5 days. Again, the actual Park phase period will depend on the time of day at which the float actually begins the next profiling cycle.

APPENDIX D: Argos Telemetry

Figure D-1: **APEX Float with Argos Telemetry**

An APEX profiling float can optionally include an Argos telemetry link as shown in Figure D-1. Argos is a satellite based system which collects, processes and disseminates environmental data from fixed and mobile platforms worldwide. During a profiling cycle, the float collects science data along with event and status information, all of which are placed into multiple 31-byte messages. Because the float does not have a receiver, it cannot determine when an Argos satellite is within range for receiving transmissions. Therefore, when at the surface, the float repeatedly transmits these messages for an amount of time at intervals of typically 40–90 seconds to ensure all the messages are received over multiple satellite passes.

NOTE *For instructions on how to run a script to convert the Argos messages to readable text, refer to “Converting the Argos Messages to Text” on page D-15.*

D.1 Argos Configuration Files

An APEX profiling float with Argos telemetry includes four configuration files:

- mission.cfg
- system.cfg
- sensors.cfg
- sample.cfg

Each file contains configuration parameters with factory default settings. The mission.cfg and sample.cfg files can be changed by the operator; the system.cfg and sensors.cfg files are factory set and should not be changed. Furthermore, files can only be changed using a PC connected directly to the float. The files cannot be changed remotely.

D.1.1 Argos system.cfg File

With the exception of the Iridium RUDICS/PSTN account information, which does not apply to an APEX profiling float with Argos telemetry, the Argos system.cfg file includes the same configuration parameters described in “system.cfg File” on page 4-14 plus the following additional ones:

argos_hex_id. The Argos Id in hexadecimal as provided by the Argos service provider.

Syntax: argos_hex_id <*Argos Id in hexadecimal*>

argos_decimal_id. The Argos Id in decimal as provided by the Argos service provider.

Syntax: argos_decimal_id <*Argos Id in decimal*>

argos_frequency. The transmission frequency of the Argos transmitter.

Syntax: argos_frequency <MHz>

The Argos system.cfg file configuration parameter settings are factory set and should not be changed by the operator.

D.1.2 Argos sensors.cfg File

The Argos sensors.cfg file is composed of the following configuration parameters:

sensor. Refer to “sensors.cfg File” on page 4-16 for information on this configuration parameter.

transmitter. Specifies the name of the Argos transmitter. The name is used to define its internal hardware configuration.

Syntax: transmitter <*transmitter_name*>
Default: telonics_st21_200a

Example: transmitter telonics_st21_200a

D.1.3 Argos mission.cfg File

The Argos mission.cfg file includes the same configuration parameters described in SECTION 3: “Operating Modes.” However, **TelemetryInterval**, as it applies to an APEX float with Argos telemetry, functions differently from that of an APEX profiling float with RUDICS/PSTN or SBD communications.

TelemetryInterval. The time in seconds between each wakeup for the message transmission of the collected science data and event and status information. Data message transmissions will continue until the end of the period specified by **UpTime**. Should the float go into Emergency mode as described in “Emergency Mode” on page

D-13, *only* the Argos Emergency Message will be transmitted each time, and for a period of four hours.

Syntax:	TelemetryInterval *<seconds>*
Range:	20–120 sec
Default:	45 sec

D.1.4 Argos sample.cfg File

Sampling for an APEX profiling float with Argos telemetry is controlled with entries in the sample.cfg file. The maximum number of samples is limited to 128 for each ascent, and *only* interval sampling can be performed. The float cannot perform continuous profiling. The specified start and stop pressures can be in any order. In addition, sampling should only be specified for the Ascent phase. The sampling behavior for the float is specified using a single entry in the sample.cfg file for each of any number of pressure ranges in accordance with the following syntax:

Syntax: ASCENT
Sample CTD *[<start> [<stop> [<interval> [<units> [<count>]]]]]*

For example, to sample the CTD every 10 dbar beginning at 1000 dbar and ending at 200 dbar, enter:

```
ASCENT
Sample CTD 1000 200 10 dbar
```

Sampling should not be specified for the Park phase, as the float will perform statistical sampling of pressure and temperature during the Park phase at the sampling interval specified by **ParkTimerInterval**.

D.2 Argos Mission Prelude Phase

At the start of the Mission Prelude phase, an APEX profiling float with Argos telemetry sends 6 test messages, one every 8 seconds. During the Mission Prelude phase, the float repeatedly transmits two 31-byte test messages, Test Message 1 and Test Message 2, where the time between each message transmission is specified by **TelemetryInterval**. The test messages are transmitted continuously for the period specified by **PreludeTime** of the Mission Prelude phase. A description of the contents of Test Message 1 is shown in Table D-1 followed by an example output, and for Test Message 2, in Table D-2 on page D-5, also followed by an example output.

Table D-1: Test Message 1

Bytes	Mnemonic	Description
0	CRC	The CRC for this message
1	MSG	The identifier number for this message
2	BLK	The block identifier for a group of transmitted messages
3	MAJOR	The major firmware revision number
4	MINOR	The minor firmware revision number
5	PATCH	The patch firmware revision number
6–8	FLTID	The **Float_id** configuration parameter setting
9–10	SEC	The time in seconds since the start of the Mission Prelude phase
11–12	STATUS	The APF-11 Status
13	TELONICS	The Telonics PTT Status
14–15	P	The pressure in centibars measured once each test-message block
16	ABP	The air bladder pressure in counts measured once each test-message block
17	BAT	The quiescent battery voltage in counts measured once for each test message block
18	UP	The **Uptime** configuration parameter setting in hours
19–20	DOWN	The **DownTime** configuration parameter setting in hours
21–22	PRKP	The **ParkPressure** configuration parameter setting in decibars
23	PPP	The **ParkDescentCount** configuration parameter setting (park buoyancy position) in counts
24	NUDGE	The **BuoyancyNudge** configuration parameter setting (during the Ascent phase) in counts
25	OK	The **MinVacuum** configuration parameter setting (internal vacuum threshold) in counts
26	ASCEND	The **AscentTimeout** configuration parameter setting in hours
27	TBP	The **air_bladder_max** configuration parameter setting in counts
28–29	TP	The **DeepDescentPressure** configuration parameter setting (profile pressure) in decibars
30	TPP	The **DeepDescentCount** configuration parameter setting (profile buoyancy position) in counts
31		Not used

The following is an example of a decoded Test Message 1:

```
*** TEST MESSAGE 1 ***
Block ID: 9
Firmware Revision: 2.7.0
Float ID: 777-0315
Seconds Since Start of Prelude: 1655
Float Status Bits: 0b00000001000000
Prelude message
Telonics Status Bits: 0b000000
Surface Pressure @ Telemetry (dbar): 0.000
Air Bladder Pressure (dbar): ~12.30
Quiescent Battery Voltage (V): ~15.12
Configured Up Time (hrs): ~21
Configured Down Time (hrs): ~219
Configured Park Pressure (dbar): 200
Configured Park Piston Position (counts): ~2176
Configured Ascent Buoyancy Nudge (counts): ~160
Configured Min Vacuum (dbar): ~9.00
Configured Ascent Timeout (hrs): ~10.00
Configured Max Air Bladder Pressure (dbar): ~11.974
Configured Profile Pressure (dbar): 2000
Configured Profile Piston Position (counts): ~256
```

In the example above, approximate values due to conversions are indicated with a "~."

Table D-2: Test Message 2

Bytes	Mnemonic	Description
0	CRC	The CRC for this message
1	MSG	The identifier number for this message
2	BLK	The block identifier for a group of transmitted messages
3	MAJOR	The major firmware revision number
4	MINOR	The minor firmware revision number
5	PATCH	The patch firmware revision number
6	FEXT	The **buoyancy_pump_max** configuration parameter setting (piston full extension) in counts
7	FRET	The **buoyancy_pump_min** configuration parameter setting (piston full retraction) in counts
8	IBN	The **InitialBuoyancyNudge** configuration parameter setting (initial

		buoyancy nudge to start the Ascent phase) in counts
9	N	The **PnPCycleLen** configuration parameter setting (park and profile cycle length)
10	PACT	The **MactivationCount** configuration parameter setting (pressure activation buoyancy position) in counts
11	DPDP	The **DeepDescentTimeout** configuration parameter setting (Deep Descent phase period) in hours
12	PDP	The **ParkDescentTimeout** configuration parameter setting (Park Descent phase period) in hours
13	PRE	The **PreludeTime** configuration parameter setting (mission prelude period) in hours
14	REP	The **TelemetryInterval** configuration parameter setting (ARGOS transmission repetition period) in seconds
15–18	ARGOSID	The Argos hex Id.
19	ARGOSFREQ	The Argos transmit frequency expressed as the number of Kilohertz away from the center frequency of 401.650 MHz.
20–21	SBESN	The serial number of the CTD
22–23	SBEFW	The CTD firmware revision number
24–27	EPOCH	The current system (GMT) time.
28–29	DEBUG	The debug verbosity
30	VAC	The internal vacuum reading in counts.
31		Not used

The following is an example of a decoded Test Message 2:

```
*** TEST MESSAGE 2 ***
Block ID: 5
Firmware Revision: 2.11.2
Piston Full Extension (counts): ~3728
Piston Full Retraction (counts): ~144
Initial Ascent Buoyancy Nudge (counts): ~304
PnP Cycle Length: 1
Pressure Activation Piston Position (counts): ~2000
Deep Profile Descent Period (hrs): 2
Park Descent Period (hrs): 2
Prelude Period (hrs): 6
Telemetry Repetition Period (secs): 44
Argos Hex ID: 0x03281298
```

```
Argos Frequency: 401.650
CTD is an RBR with firmware revision 1.300
CTD Serial #: 040376
GMT time: 2018-05-01T17:57:10Z
Log Verbosity: 5
Internal Vacuum: ~10.39
```

In the example above, approximate values due to conversions are indicated with a “~.”

D.3 Argos Surface Phase

During the Surface phase for an APEX profiling float with Argos telemetry, the float repeatedly transmits up to twenty-nine 31-byte data messages, Data Message 1 to Data Message 29, where the time between each message is specified by **TelemetryInterval**. The messages are transmitted continuously from the beginning of telemetry during the Surface Phase until the expiration of **UpTime**. Descriptions of the contents of Data Message 1 through Data Message 4 are shown in Table D-3 through Table D-6, respectively, each followed by an example output. In the example outputs, approximate values due to conversions are indicated with a “~.” Data Message 5 through Data Message 29 are structured similar to that of Data Message 4.

Table D-3: Data Message 1

Bytes	Mnemonic	Description
0	CRC	CRC for this message
1	MSG	Identifier number for this message
2	BLK	Block identifier for a group of transmitted messages
3–5	FLT	The **Float_id** configuration parameter setting
6	PRF	The current mission profile number (wraps to 0 from 255)
7	LEN	The number of PTS samples in this message block
8–9	STATUS	The APF-11 Status
10	TELONICS	The Telonics PTT Status
11	CP	The current pressure in centibars as recorded during the creation of each Argos message block. Each distinct copy of Argos message #1 contains a new pressure measurement.
12–13	SP	The surface pressure in centibars as recorded just prior to the descent to the park depth.
14	SPP	The buoyancy position in counts recorded when the surface-detection algorithm terminated.

15	PPP2	The buoyancy position in counts recorded at time that the park phase of the profiling cycle terminated.
16	PPP	The buoyancy position in counts recorded at the time that the last deep-descent phase terminated.
17–20	CTD	The CTD Status
21	VQ	The quiescent battery voltage in counts measured when the park phase of the profile cycle terminated.
22	IQ	The quiescent battery current in counts measured when the park phase of the profile cycle terminated.
23	VCTD	The battery voltage in counts measured when the CTD sampled after the park phase of the profile cycle terminated.
24	ICTD	The CTD current in counts measured when the CTD sampled after the park phase of the profile cycle terminated.
25	VHPP	The average battery voltage in counts measured during the initial extension of the buoyancy pump at the start of the profile phase of the profile cycle.
26	IHPP	The average buoy pump current in counts measured during the initial extension of the buoyancy pump at the start of the profile phase of the profile cycle.
27	VAP	The average battery voltage in counts measured during the most recent period when the air pump was activated.
28	IAP	The average air pump current in counts measured during the most recent period when the air pump was activated
29–30	VSAP	The integrated measure (Volt-Sec) of the volume of air pumped during the telemetry cycle.
31	NA	Not used.

The following is an example of a decoded Data Message 1:

```
*** DATA MESSAGES DATA START ***
Float ID: 777-0315
Profile #: 1
Block ID: 7
# of Profile Samples: 100
Float Status Bits: 0b00000000000001
Deep Profile occurred
Telonics Status Bits: 0b000000
Pressure (dbar): 0.000
Surface Pressure @ Park Descent Start (dbar): 0.000
```

```
Piston Position @ Surface Detection (counts): ~3680
Piston Position @ Park End (counts): ~2624
Piston Position @ Deep Descent End (counts): ~256
CTD Status Bits: 0b00000000000000000000000000000000
Quiescent Battery Voltage @ Park End (V): ~15.12
Quiescent Battery Current @ Park End (cnts): ~255
Quiescent Battery Current @ Park End (mA): ~19.92
Battery Voltage @ Park End With CTD Powered (V): ~0.00
CTD Current @ Park End With CTD Powered (mA): ~0.000
Average Battery Voltage During Initial Ascent Nudge (V): ~15.05
Average Buoy Pump Current During Initial Ascent Nudge (cnts):
~416
Average Buoy Pump Current During Initial Ascent Nudge (mA):
~195.786
Average Battery Voltage During Air Pump Activation (V): ~15.05
Average Air Pump Current During Air Pump Activation (cnts):
~800
Average Air Pump Current During Air Pump Activation (mA):
~97.656
Volt-Sec for Air Pumped During Telemetry: 603
```

In the example above, approximate values due to conversions are indicated with a "~."

Table D-4: Data Message 2

Bytes	Mnemonic	Description
0	CRC	CRC for this message
1	MSG	Identifier number for this message
2 - 5	EPOCH	Unix epoch when down time expired.
6–7	TINIT	Time (minutes) when telemetry phase was initiated relative to EPOCH. (2's complement signed integer)
8	NADJ	The number of active-ballast adjustments made during the park phase
9–10	PRKN	The number of hourly park-level PT samples.
11–12	TMEAN	The mean temperature of park-level PT samples.
13–14	PMEAN	The mean of pressure differences of park-level PT samples.
15–16	SDT	The standard deviation of temperature of park-level PT samples.
17–18	SDP	The standard deviation of pressure of park-level PT samples.
19–20	TMIN	The minimum temperature of park-level PT samples.

21–22	TMINP	The pressure associated with TMIN of park-level PT samples.
23–24	TMAX	The maximum temperature of park-level PT samples.
25–26	TMAXP	The pressure associated with TMAX of park-level PT samples.
27–28	PMIN	The minimum pressure of park-level PT samples.
29–30	PMAX	The maximum pressure of park-level PT samples.
31	NA	Not used.

The following is an example of a decoded Data Message 2:

```
Downtime Expiration Time: 2017-04-23T16:35:46Z
Telemetry Start Downtime End Delta (mins): 376
# of Ballast Adjustments: 46
# of Park Level PT Samples: 211
Mean Temperature for Park Level PT Samples: 4.406
Mean Park Pressure Diff for Park Level PT Samples: 138.900
Std Dev of Temperature for Park Level PT Samples: 0.002
Std Dev of Park Pressure Diff for Park Level PT Samples:
128.700
Min Temperature for Park Level PT Samples: 4.405
Pressure @ Min Temperature for Park Level PT Samples: 358.000
Max Temperature for Park Level PT Samples: 4.406
Pressure @ Max Temperature for Park Level PT Samples: 222.100
Min Pressure for Park Level PT Samples: 221.700
Max Pressure for Park Level PT Samples: 602.200
```

Table D-5: Data Message 3

Bytes	Mnemonic	Description
0	CRC	CRC for this message
1	MSG	Identifier number for this message
2	VAC	The internal vacuum in counts recorded when the park phase of the profiling cycle terminated.
3	ABP	The air bladder pressure in counts recorded just after each Argos transmission.
4	PAP	The number of 5-second pulses of the air pump required to inflate the air bladder
5 - 6	PMT	The total length of time in seconds that the pump motor ran during the current profile cycle.

7–8		Temperature
9–10		Pressure
11–12		Salinity
13–14		Temperature
15–16		Pressure
17–18		Salinity
19–20		Temperature
21–22		Pressure
23–24		Salinity
25–26		Temperature
27–28		Pressure
29–30		Salinity
31	NA	Not used.

The following is an example of a decoded Data Message 3:

```
Internal Vacuum @ Park End (dbar): ~10.39
Air Bladder Pressure @ End of Air Activation (dbar): ~0.00
# of 5 Second Air Pump Pulses: 8
Run-time for Buoy Pump (secs): 88
*** Sample Data Start ***
Temperature, Pressure, Salinity
4.406, 1999.000, 16.613
4.406, 1949.300, 16.625
4.406, 1898.500, 16.637
4.406, 1849.100, 16.649
4.406, 1799.300, 16.662
4.406, 1749.300, 16.674
4.406, 1699.400, 16.687
4.406, 1648.600, 16.699
```

In the example above, approximate values due to conversions are indicated with a “~.”

Table D-6: Data Message 4

Bytes	Mnemonic	Description
0	CRC	Message CRC
1	MSG	Message ID
2–3		Temperature
4–5		Pressure
6–7		Salinity
8–9		Temperature
10–11		Pressure
12–13		Salinity
14–15		Temperature
16–17		Pressure
18–19		Salinity
20–21		Temperature
22–23		Pressure
24–25		Salinity
26–27		Temperature
28–29		Pressure
30		Salinity (byte 1, where byte 2 is contained in Data Message 5)
31		Not used.

The remaining available messages for sampling data are Data Message 5 to Data Message 29, each containing a Message CRC, a Message ID, and temperature, pressure, and salinity data. Bytes 21–30 for Data Message 29 are not used. However, depending on the float configuration and sampling behavior there may be less messages transmitted.

D.4 Float Status Information

Float status bits are sent in Test Message 1 as STATUS bytes 11–12 and in Data Message 1 as STATUS bytes 8–9 as follows with Bit 1 as the rightmost bit:

Bit 1:	Deep profile first
Bit 2:	Float bottom detection
Bit 3:	Sample timeout
Bit 4:	Piston fully extended
Bit 5:	Ascent timeout occurred
Bit 6:	Test message
Bit 7:	Prelude message
Bit 8:	(future) Ice breakup detection
Bit 9:	Emergency flag set
Bit 10:	CTD exception
Bit 11:	CTD unreliable
Bit 12:	(future) Ice detection
Bit 13:	Telecommunications exception
Bit 14:	Air system bypass
Bit 15:	Watchdog detection
Bit 16:	Profile ID overflow

D.5 Emergency Mode

An APEX profiling float with Argos telemetry will go into Emergency mode if a catastrophic error occurs during any mission phase or if the float fails the self test during the Mission Prelude phase. When in Emergency mode, the float will continuously transmit *only* the Argos Emergency Message for four hours, where the time between each message is specified by **TelemetryInterval**, after which the Emergency mode will be automatically cleared. A description of the contents of the Argos Emergency Message is shown in Table D-7. Once cleared the float will again attempt to run the mission. If the float again goes into Emergency mode, this cycle is repeated until the batteries are depleted.

Table D-7: Argos Emergency Message

Bytes	Mnemonic	Description
0	CRC	CRC for this message
1	MSG	Identifier number for this message
2	BLK	Block identifier for a group of transmitted messages
3	MAJOR	Major firmware revision number
4	MINOR	Minor firmware revision number
5	PATCH	Patch firmware revision number
6 - 8	FLT	The **Float_id** configuration parameter setting
9	TELONICS	The Telonics PTT Status
10	ABP	The air bladder pressure in counts.
11	BAT	The Quiescent battery voltage in counts
12	VAC	The internal vacuum reading in counts.
13-16	PC	The exception program counter.
17	VAP	The average battery voltage in counts measured during the most recent period when the air pump was activated.
18	IAP	The average battery current in counts measured during the most recent period when the air pump was activated
19 – 20	VSAP	The integrated measure (Volt-Sec) of the volume of air pumped during the telemetry cycle.
21-31	NA	Not used

D.6 Converting the Argos Messages to Text Files

The Argos 31-byte message files for each surfacing must first be obtained from the Argos service provider and then converted to text files before they can be read. This operation requires that you first download and install Python on a PC. To install Python, go to http://www.python.org/downloads/ and download and install Version 2.7.11 in the C:\ directory of your PC. In addition, the apf11_argos_msgs_decoder.py Message Decoder script, provided by Teledyne Webb Research, must also be saved in the C:\ directory of the PC. This script processes files in two different formats. One format is for factory use only, and the other format is expected to be used by operators.

The Python script apf11_argos_msgs_decoder.py Message Decoder script processes message files that can include test and data messages. The script can process message files in a few different formats using program options to indicate the input file format. In most cases the file downloaded from the Argos service provider may have a header line. The python script command includes an “-h” option to process the file header. In addition, the data fields in the input data files are typically separated by a comma. In this case the “-s” option followed by a comma in quotes should be used to indicate that it is the field separation character. If the input data file uses a different field separation character, the “-s” option must be followed by the desired separation character in quotes.

The Message Decoder command is the following:

Syntax: *python apf11_argos_msgs_decoder.py [-h] [-s <field_delimiter>] <input-file-name> [> <output-file-name>]*

-h: Indicates that the input data file has a header line to process before the message data, for example “SENSOR 01,” “SENSOR 02,” etc.

-s <field_delimiter>: Indicates the delimiter used to separate message record fields in the input data file, such as a comma.

<input-file-name>: The name of the file for the messages to be processed. The name may include the directory path.

<output-file-name>: The name of the file to redirect output from the python script. The name may include the directory path.

To convert the Argos messages to text files:

1. Log onto the service provider site and enter the information on the form, including the following:
   - Select “By Platform ID Numbers.”
   - Enter the decimal Id for float.
   - Select the time frame for message capture associated with the float’s surfacing.

TELEDYNE WEBB RESEARCH
Everywhereyoulook™

- Select field identifiers for SENSOR 01 through SENSOR 31.
- Select CSV output format.

2. Download the Argos message files from the Argos service provider into a directory of the PC.
3. Enter the Message Decoder command in accordance with the syntax.
4. Open the output file in a text display program, such as Microsoft Notepad.

To convert Argos messages from parseXml files to text files:

1. Copy the original parseXml files to a folder. Name the folder ParseXMLFiles for example.
2. Create a folder to contain the generated timestamped files. Name the folder TimestampedFiles for example.
3. Determine the Argos decimal-id for the parseXml files to be processed. For example, in the example to follow, it is 052918.
4. Run the following script using the above named folders and decimal-id:

   ```
   python generate_timestamped_files_from_email_format.py –d
   052918 –i ParseXMLFiles –o TimestampedFiles –g parseXml
   ```

   This process generates "timestamped" files from the parseXml files.
5. Calculate the **UpTime** minus **AscentTimeout** in hours for the float from which the data originated. This time is used to divide the processed data into different profiles. For this example float, **UpTime** – **AscentTimeout** = 8 hours.
6. Run the following script using the above named folders and decimal-id:

   ```
   python organize_msg_files.py –i TimestampedFiles –o
   OrganizedFiles –h 8
   ```

   This process generates "organized" files from the parseXml files.
7. Check the separator used in the organized files. In this example it is a comma (“,”).
8. Run the following script on the list of files in the organized files folder:

   ```
   python apf11_argos_msgs_decoder.py –s “,” <list of files
   separated by a space>
   ```

   This process runs the decoder on the organized files. If using Linux you can use “*” to indicate all files. If using Microsoft Windows, you may need to list all the files separately.

D.7 Conversion from Hexadecimal to Physical Units

The pressure, temperature, salinity, battery voltage, battery current, air bladder pressure, and other values measured by the float are encoded in the data messages as hex integers. This compression reduces the number of bytes in the Argos transmissions, but also reduces the resolution of the encoded data. For example, battery current in mA is represented by a single 8-bit byte, which can represent integers 0 through 256. To increase the size, but not the resolution, of the encoding, the value is scaled from an 8-bit value to a 12-bit value by multiplying by 2 to the power 4, or 16. This adds another 4 bits to the representation. The table below illustrates this with a transmitted hexadecimal battery current count of 0x10, namely 16, being multiplied by 16 to give a *raw* value of 256 before being converted to physical units of mA by the 0.07813 multiplier.

To convert hex values in Argos messages back to physical units, proceed as illustrated in the examples shown in Table D-8. The initial conversion from hexadecimal to decimal should assume the hex value is an unsigned integer with a range (for 2-byte values) of 0 to 65535 for pressure, temperature and salinity. Negative pressures can be represented with an initial hex value greater than 32767 (equivalent to 0x7FFF). Negative temperatures cannot currently be represented, although this is a known issue that is being addressed. Other values, such as battery voltage and current, etc. are represented by 1-byte values. In Table D-8, raw values (such as Praw) represent conversion to decimal before being converted to physical values, such as pressure in dbar or temperature in °C.

Table D-8: Argos Data Message Hexadecimal to Decimal Conversions

Measurement	Hexadecimal	Decimal and Conversion Steps	Resulting Physical Value
Pressure > 0	0x0397 (<= 0x7FFF)	P_{raw} = 919 P = P_{raw} / 10.0	91.9 dbar
Pressure < 0	0xFFF9 (> 0x7FFF)	P_{raw} = 65529 P = -1x(65535 – P_{raw})/10.0	-0.6 dbar
Temperature > 0	0x1135	T_{raw} = 4405 T = T_{raw} / 1000.0	4.405 °C
Salinity > 0	0x42DC	S_{raw} = 17116 S = S_{raw} / 1000.0	17.116 PSU
Battery volt counts	0xD6	V_{raw}= 214 V = V_{raw} * 16 * (18.0/4096.0)	~15.05 V

Air bladder pressure counts	0x64	Praw = 100 P = Praw * 16 * (20.852/4096.0)	~8.15 dbar
Vacuum counts	0xFF	Vraw = 255 V= Vraw * 16 * (10.4302/4096.0)	~10.39 dbar
Buoy pump current counts	0x1A	Iraw = 26 I = (Iraw-19.98)*(2025.0/4096.0)	~195.78 mA
Air pump current counts	0x32	Iraw = 50 I = Iraw * 16 * (500.0/4096.0)	~97.656 mA
Battery current counts	0x10	Iraw = 16 I = Iraw *16 * 0.07813	~20.00 mA

In the table above, approximate values due to conversions are indicated with a "~."

APPENDIX E: Hyper Retract/N2 Compensation

The hyper retract feature can be used with or without N2 compensation hardware on a standard APEX profiling float. With N2 compensation, hyper retraction is typically required. However, hyper retraction by itself does not require N2 compensation. With the N2 compensator installed in a standard APEX profiling float, the piston is fully retracted and nitrogen is compressed by external pressure on the oil compartment of the rubber bladder. When used independently, hyper retraction pulls in an amount of oil to temporarily increase the float density as it leaves the surface. The increased float density causes the float to descend at a faster than usual rate. The hyper retraction feature requires both the specification of a buoyancy position for adjusting the oil bladder volume and the pressure at which to reset the buoyancy position back to **ParkDescentCount**.

E.1 Configuring Hyper Retraction

Two mission.cfg parameters are provided for configuring hyper retraction: **HyperRetractCount** and **HyperRetractPressure**. **HyperRetractCount** is the buoyancy position in counts when the float leaves the surface and begins the Park Descent phase. When set to a nonzero value, hyper retraction is enabled. **HyperRetractPressure** is the pressure at which the buoyancy position is set back to **ParkDescentCount** when **HyperRetractCount** is enabled. For more information on **HyperRetractCount** and **HyperRetractPressure**, refer to “Park Descent Phase” on page 3-6.

When hyper retraction *is not* enabled, the float moves the buoyancy position to **ParkDescentCount** at the start of the Park Descent phase, after the air bladder is deflated.

When hyper retraction *is* enabled, the buoyancy position is set to **HyperRetractCount** at the start of Park Descent. This count value is typically set to a lower value than **ParkDescentCount**. When the float reaches **HyperRetractPressure**, the buoyancy position is adjusted to **ParkDescentCount**.

In the following examples, entries are made in the mission.cfg file.

Example 1. To enable hyper retract and move the buoyancy position to 500, enter:

```
HyperRetractCount 500
```

Example 2. To move the buoyancy position to **ParkDescentCount** when the float reaches 900 dbar, enter:

```
HyperRetractPressure 900
```

E.2 Usage Considerations

There are several scenarios to consider when enabling hyper retraction:

- The float reaches **HyperRetractPressure**.
- The float reaches **ParkPressure**–**ParkDeadBand**.
- Float reaches **Park Descent Timeout** during the Park Descent phase.

The normal case is when the float reaches **HyperRetractPressure** before **ParkPressure**–**ParkDeadBand** or **ParkDescentTimeout**. In this case the float moves the buoyancy position to **ParkDescentCount**. If **ParkDescentTimeout** occurs during hyper retraction, the float will immediately reset the buoyancy position to **ParkDescentCount**, and potentially ascend to a shallower depth. If the float reaches **ParkPressure**–**ParkDeadBand** during hyper retraction, the float will immediately reset the buoyancy position to **ParkDescentCount**. For more information on **HyperRetractPressure** and **HyperRetractCount**, refer to "Park Descent Phase" on page 3-6.

NOTE ***ParkDescentTimerInterval*** *should be set accordingly so that the float wakes up to detect* ***HyperRetractPressure*** *before the float reaches the Park depth.*

NOTE *The float takes several seconds to move oil from* ***HyperRetractCount*** *to the* ***ParkDescentCount****. There will be a lag time between when the float starts traveling at the rate associated with the* ***ParkDescentCount*** *versus the* ***HyperRetractCount*** *that should be considered when determining* ***HyperRetractPressure*** *relative to* ***ParkPressure****.*

E.3 N2 Compensator

The N2 Compensator is optional hardware installed as part of the standard APEX float buoyancy system to enable profiling from 2000 decibars through very low density surface conditions. While the float is at the surface, nitrogen gas in the compensator expands adding approximately 80 grams of buoyancy to the float. During the Park Descent phase, the buoyancy position is moved to **HyperRetractCount**, which is typically the minimum possible value, and the gas expands to fill the available space. As external pressure increases during descent, the gas is compressed, reaching an effective minimum at approximately 700 meters. Upon reaching **HyperRetractPressure**, the buoyancy position is set to **ParkDescentCount**, and the float will settle in near **ParkPressure.** This allows floats with an N2 compensator to park at pressures greater than 850 dbar.

CAUTION *The N2 Compensator renders the float buoyantly unstable in the range of pressures of approximately 400–700 decibars. Therefore such floats must be at a Park depth that is outside this range.*

APPENDIX F: Setting up Communications

The APEX profiling float behaves in accordance with a factory programmed mission that can be modified by the operator either locally by connecting the float to a PC or remotely over the Iridium satellite network after the float is deployed. A mission is modified by changing the settings of the configuration parameters in the mission.cfg file. These configuration parameters are downloaded by the float during the Mission Prelude phase and during each Surface phase. In addition, a number of user commands are available that can be used to control some of the float functions and access its files when it is connected to a PC.

NOTE *In addition to the mission plan, the sampling behavior of the float can be modified remotely by changing the settings of the configuration parameters in the sample.cfg file. Similar to that for a mission.cfg file, these settings are downloaded by the float during the Mission Prelude phase and during each Surface phase. The settings can also be edited locally by connecting a float to a PC. For instructions on how to edit these configuration parameter settings locally, refer to "Viewing and Editing the Configuration Parameters" on page 2-14.*

F.1 Setting up RUDICS/PSTN Communications with the Float

To receive data from a deployed float and to be able modify the float's mission.cfg file remotely, you can set up a server and either connect it to an Iridium gateway using an Iridium modem with a dial-up connection to the Public Switched Telephone Network (PSTN), or use Router-Based Unrestricted Digital Internetworking Connectivity Solution (RUDICS) which enables communications with the float over the internet. For redundancy you can set up both systems, each with its own server, or use one server for both with RUDICS being the primary communications path and PSTN being the secondary. In addition, a PC with an internet connection is required. Use of the PSTN incurs a low setup cost but higher data transfer costs compared to RUDICS which has a high setup cost but lower data transfer costs. In both cases a login account must be created on the server or servers by the system administrator for the float, and a home directory must be assigned for receiving data. This directory must also include the mission.cfg file. For detailed information on how to set up either system, contact Teledyne Webb Research customer service.

NOTE *Teledyne Webb Research also provides as an optional service a Unix based host server for receiving float data and remotely modifying the mission plan for the APEX profiling floats. Other host server providers are also available. Contact Teledyne Webb Research customer service for more information.*

F.2 Setting up SBD Communications with the Float

Iridium Short Burst Data (SBD) may be used to communicate with an APEX profiling float. Typical shore side communications to and from the Iridium SBD Gateway are through e-mail. The floats use the Iridium 9602 Satellite Transceiver for SBD communications.

To receive log file data from a deployed float, and to be able modify the float's configuration files while it is deployed, you can associate up to 5 e-mail accounts with the SBD service. Log file data are transmitted *from* the float as Mobile Originated SBD (MO-SBD) messages that are 1 to 340 bytes long. Mission configuration files are transmitted to the float as Mobile Terminated SBD (MT-SBD) messages that are from 1 to 270 bytes long.

MO-SBD messages originated from the floats are delivered to the Iridium Gateway by way of the satellite network and subsequently attached as binary files to one or more e-mails for delivery to the provisioned destination e-mail addresses. Teledyne Webb Research provides a Python script described below to recombine the .sbd messages into compressed binary science_log and vitals_log files, as well as, ascii system_log files. Scripts are also provided to further decode these binary files to readable CSV and TXT files.

Updates to mission.cfg and sample.cfg files may be sent to an SBD equipped float. The MT-SBD messages for delivery to a float are sent to the Iridium Gateway (data@SBD.iridium.com) by e-mail. A single e-mail may contain several .sbd attachments, where each attachment constitutes an SBD message. Notification e-mail is sent to the originating (From:) e-mail address indicating whether messages are queued for transmission or rejected. Teledyne Webb Research provides a Python script described below that will appropriately segment the mission.cfg or sample.cfg update file, or both, into .sbd binary messages and transmit these as e-mail attachments.

F.2.1 SBD Packet Formats

The first byte of an SBD message at index 0 indicates the start of a new file or a file continuation:

- 0x01 <- Start of a new file
- 0x02 <- File continuation

If the SBD message is the start of a new file:

- The bytes at index 1 and 2 are the packet count for the entire file to be transmitted.
- The next bytes 3 thru N up to null char (i.e. \0) or max 336 bytes contain the filename in utf-8.
- The remaining bytes up to 340 maximum contain the data from the APEX float.

If the SBD message is the continuation of a file, the remaining bytes, up to 339, contain additional data from the float.

F.2.2 SBD Scripts

sbd_send.py -s <smtp-server> -u <user-email> -p <password> -i <imei> -f <filepath>

-s <smtp-server>:	Remote SMTP server IP address or hostname where an e-mail server resides that allows for SMTP over SSL connections.
-u <user-email>:	User e-mail account known by the specified e-mail server.
-p <password>:	User e-mail password credentials used to connect to the remote e-mail server.
-i <imei>:	IMEI for the SBD modem installed on the APEX float.
-f <filepath>:	Qualified path to file that should be transmitted to the APEX float.

The sbd_send.py python script is used to send files to an APEX float. This script connects to a specified remote SMTP server via SMTP over SSL and sends a specific file as one or more SBD attachments to e-mail address data@sbd.iridium.com.

sbd_imap_email_downloader.py -r <remote-imap-server> -u <username> -p <password> -i <imei> -f <filepath> [-s <sent-on>] [-n]

-r <remote-imap-server>:	The address of the remote imap server.
-u <username>:	User e-mail account.

-p <password>:	User e-mail password credentials.
-i <imei>:	IMEI for the SBD modem installed on the APEX float.
-f<filepath>:	Qualified path to folder used for files transmitted by the APEX float. If the folder path does not exist, the script will create the folder path.
-s <sent-on>:	Used to only look for e-mail received on a specific date. The date should be in the format of "dd-mmm-YYYY" where "dd" is a two digit day of the month, "mmm" is the first three characters of a month with the first character being capitalized and YYYY is a four digit year.
-n:	Indicates that the script should only download unseen e-mails.

The sbd_imap_email_downloader.py python script is used to download files transmitted by a float to an e-mail server that supports IMAP4 over SSL connections and connects to IMAP4 e-mail over SSL and download e-mails associated with a specific IMEI.

- Downloaded e-mails will be saved into the specified output path folder with the extension ".sts" and SBD e-mail attachments will be saved with the extension ".sbd."
- This script will only look for e-mails from sbdserver@sbd.iridium.com where the e-mail subject is "SBD Msg from Unit <IMEI>", where <IMEI> is the IMEI number.

sbd_data_file_processor.py -d <filepath>

-d <filepath>:	Qualified path to a folder that contains sbd data files.

The sbd_data_file_processor.py python script is used to process downloaded sbd files that were generated from e-mail attachments.

- The SBD file name is in the format "<IMEI>_<MOMSN>.sbd." Where MOMSN is the Mobile Originated Message Sequence Number.
- This script is expected to process sbd files for a specific IMEI.
- This script should not be used to process sbd files for multiple IMEIs.
- Processed files will be placed in an "archive" subfolder.

APPENDIX G: APEX-EM Electromagnetic Float

The Teledyne Webb Research APEX-EM (Autonomous Profiling Explorer) electromagnetic profiling float continually acquires water current, temperature and salinity profiles. The current velocity measurements can be used to characterize internal wave properties, upper-ocean dynamics, and patterns of geostrophic shear, even under hurricane conditions.

G.1 General Description

The APEX-EM electromagnetic profiling float is specifically designed to obtain water current velocity through motional induction as it descends and ascends in the water column. The float uses an electromagnetic current measurement system developed by the Applied Physics Laboratory of the University of Washington (APL-UW). The current velocity is obtained by measuring the motionally induced electric fields generated by ocean currents moving through the vertical component of the Earth's magnetic field. The fields are sensed using externally mounted electrodes. The electrodes output motionally induced voltages which are amplified and digitized, and the digitized data are stored in the float's internal memory. The current measurement system also includes a 3-axis magnetometer and a 3-axis accelerometer.

G.2 Construction and Main Components

The APEX-EM electromagnetic profiling float is shown in Figure G-1. Its main components are a housing, electrodes, a Preamplifier board, a Compass and Tilt board, and an EM Controller board.

G.2.1 Housing

The housing is constructed of anodized aluminum and coated with yellow epoxy paint. It includes a sixteen bladed vane which rotates the float as it descends or ascends in the water column. The rotation enables mathematical separation of the voltage measurements from the electrode drift and ensures that each electrode channel samples both components of velocity.

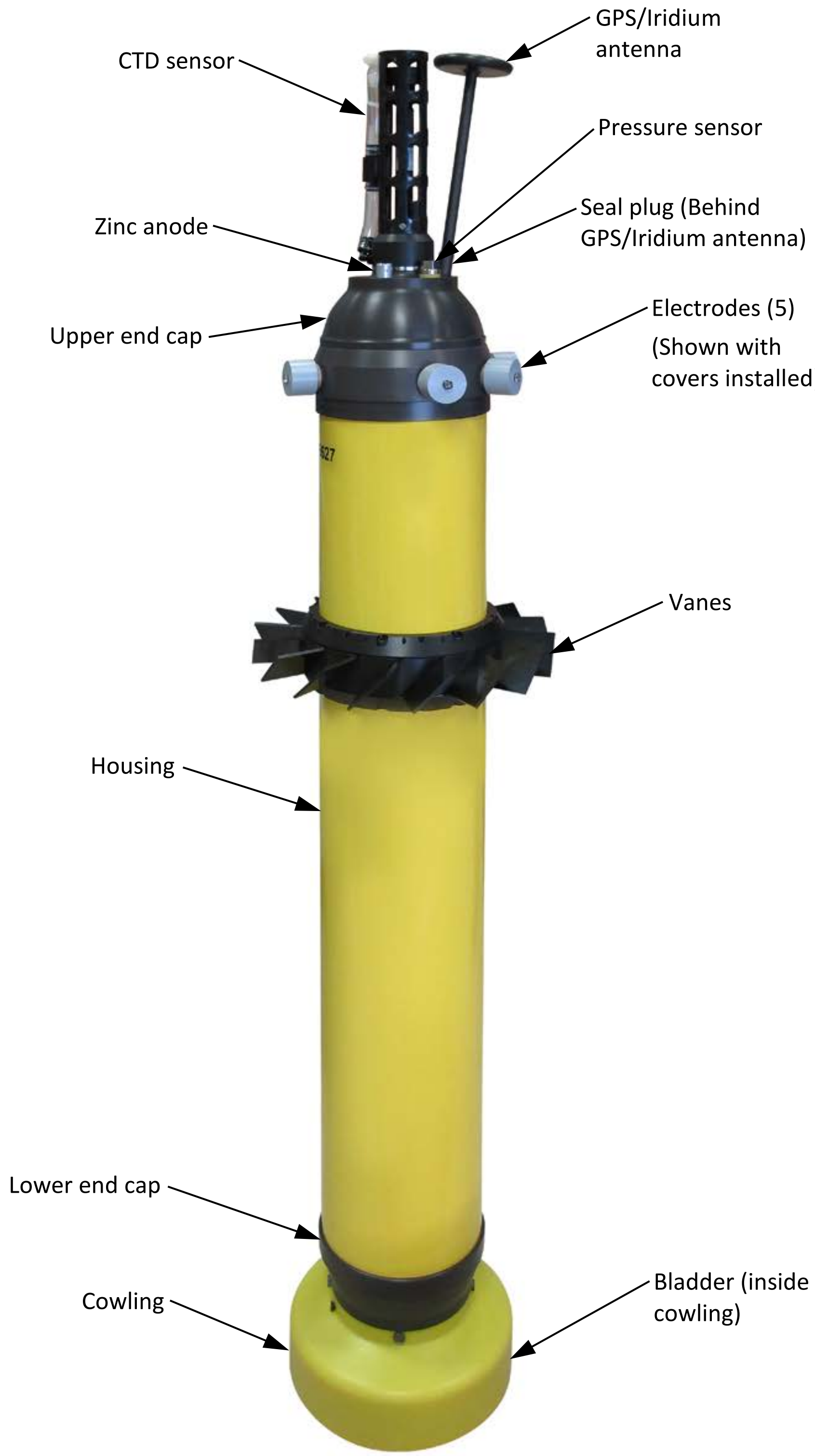


Figure G-1: The APEX-EM Electromagnetic Profiling Float Main External Components

G.2.2 Electrodes

Five Ag/AgCl (silver/silver chloride) electrodes are each housed in a chamber filled with agar. The agar is made from salt water and acts to buffer the electrodes from temperature and salinity changes. The electrodes are mounted around the outside circumference of the top end cap. Four are positioned as two orthogonal pairs, and the fifth serves as a reference for electrode preamplifiers inside the float. The electrodes measure the motionally induced electric fields generated from the motion of seawater and the vertical motion of the float.

G.2.3 Preamplifier Board

The Preamplifier board contains two low noise preamplifiers which amplify the outputs of the two orthogonal pairs of electrodes.

G.2.4 Compass and Tilt Board

The Compass and Tilt board includes a 3-axis accelerometer and a 3-axis magnetometer. The accelerometer provides two horizontal channels of orthogonal acceleration that are used to determine float tilt. The magnetometer provides three channels of orthogonal magnetic field measurements that are used with tilt to locate magnetic north.

G.2.5 EM Controller Board

The EM Controller board includes a microcontroller, a vertical accelerometer, an optically isolated serial interface, and voltage regulators. The microcontroller inputs the electric field outputs from the Preamplifier board and provides 24-bit analog to digital conversion of these outputs and 10-bit analog to digital conversion of the magnetic field and acceleration outputs from the Compass and Tilt board and the vertical accelerometer. The vertical accelerometer provides a third channel of acceleration orthogonal to the two on the Compass and Tilt board. The microcontroller also computes data averages and provides some data buffering. All the data are output to the Controller board over the optically isolated serial interface. The voltage regulators input power from the two battery packs and provide voltage regulated power to the EM Controller board itself, to the Compass and Tilt board and to the Preamplifier board.

G.3 Specifying the EM Hardware Configuration

A line in the sensors.cfg file defines the EM hardware configuration:

```
sensor apluw_ema COM4 B38400
```

For more information on the sensors.cfg file along with additional examples, refer to “sensors.cfg File” on page 4-16.

G.4 Specifying the EM Air Bladder Behavior

The EM float requires minimum descent and ascent rates so that the float can properly spin as it collects data from the EM sensors. To maximize the descent rate the float can be configured to keep the air bladder inflated at the surface while it retracts oil to prepare for the Park Descent. After the oil is pulled in, the float will deflate the air bladder. In this case, the **ParkDescentTimeout** should be modified accordingly to include the time needed to adjust the buoyancy position.

The mission.cfg parameter for controlling the air bladder deflation behavior is **ParkDescentAirAfter**.

ParkDescentAirAfter. If set to "on," the float will keep the air bladder inflated while it retracts oil at the surface for the Park Descent phase. If set to "off," the float will deflate the air bladder at the surface before retracting oil. When set to “off,” which is the default, this configuration parameter becomes hidden and therefore is not displayed in the log file output of the mission parameters.

Syntax:	ParkDescentAirAfter <on|off>
Range:	on or off
Default:	off

G.5 Specifying the EM Mission Parameters

The EM float includes several mission parameters that are configured to specify the EM float mission behavior. The EM float needs to travel at a minimum rate for EM data collection during descent and ascent.

The mission.cfg file configuration parameters that apply to the EM float mission behavior are **AscentRate**, **HyperRetractCount**, **HyperRetractPressure, ParkDescentTimeout, ParkDescentAirAfter** and **ParkDescentCount**.

The **AscentRate** configuration parameter is used to control the ascent rate of the EM float. It is typically set to 0.12 dbar/sec. **HyperRetractCount**, **HyperRetractPressure** and **ParkDescentTimeout** are used to control the descent rate of the float. Typically, the rate will increase as more oil is retracted. **ParkDescentAirAfter** described above is used to hold the float at the surface while the oil is retracted. **ParkDescentCount** should be chosen to be the neutrally buoyant position for **ParkPressure**. This setting will optimize the float's turn around time when reaching the Park phase.

The EM float typically runs a saw tooth pattern for EM data collection where it spends no time in the Park phase. **DownTime** is set accordingly to minimize the time in the Park phase.

G.6 Specifying the EM Configuration

The EM float has several parameters that can be configured to specify EM data collection behavior. The parameter values are provided in each of the science logs sent by the float.

ema_config_set. Sets a specified EM configuration parameter to a specified setting.

Syntax: `ema_config_set <Configuration Parameter Name> <Value>`

ema_config_get. Displays a specified EM configuration parameter.

Syntax: `ema_config_get [<Configuration Parameter Name>]`

The EM float configuration parameters that apply to the EM data collection are **EMALogMissionCycle**, **EMALogMaxPressure**, **ReqSamples,** and **SlideSamples**:

EMALogMissionCycle. Specifies how often raw EM data should be collected in an ema_log file. A setting of 0 disables raw EM data collection. A value of 1 is every mission cycle, a value of 2 is every other mission cycle. The default is every 10th mission cycle.

Syntax: ema_config_set **EMALogMissionCycle** <*Value*>
Range: 0, 1–255
Default: 10

EMALogMaxPressure. Specifies the maximum pressure to record raw EM data in an ema_log file. The raw data is collected for both the descent and ascent phases.

Syntax: ema_config_set **EMALogMaxPressure** <*Value*>
Range: 0–2500 dbar
Default: 100 dbar

ReqSamples. Specifies the required number of raw 1hz samples to collect for creating one processed entry in the science log. A value of 50 specifies that 50 collected samples are processed to create an EM data log entry.

TELEDYNE WEBB RESEARCH Everywhereyoulook™

Syntax: ema_config_set **ReqSamples** <*Value*>

Range: 10–70

Default: 50

SlideSamples. Specifies the number of older raw 1hz samples to remove from the **ReqSamples** data set for obtaining newer samples to creating the next processed entry in the science log. A value of 25 specifies that the oldest 25 collected samples are removed to refresh with new samples to process with the remaining samples to create an EM data log entry.

Syntax: ema_config_set **SlideSamples** <*Value*>

Range: 10–70

Default: 25

The EM configuration parameters described above are displayed in the science log. They are also displayed in the float's self test output for the EM sensor. The row in the science_log.csv file with the first column label "EMA_CFG" contain EM configuration data. The columns are from left to right:

Sensor type: EMA_CFG

Time stamp: UTC (YYYYMMDDTHHMMSS)

Required EM Data Samples: The number of raw 1 Hz EM samples collected for the science log entry

EM Data Collection Window Slide: The number of EM samples removed from the data set for refresh.

EM Raw Data Collection Max Pressure: The maximum pressure used to log raw EM data

EM Raw Data Mission Cycle: The mission cycle frequency for reporting raw EM data

An example row of EM configuration data extracted from a science_log.csv file is shown in Figure G-2.

EMA_CFG	20200521T173508	50	25	100	10

Figure G-2: Example Row of EM Configuration Data

In this example, 50 EM data samples are collected and processed for a science log entry. There is a sliding window of 25 samples used to remove the oldest samples and refresh with new samples for processing. Raw data is collected above 100 decibars during float descent and ascent for every 10^{th} mission cycle.

G.7 Specifying the EM Sampling Behavior

The interval sampling behavior for the EM float is specified using a single entry in the sample.cfg file for each of any number of pressure ranges in accordance with the following syntax:

<phase>
SAMPLE EM *[<start> [<stop> [<interval> [<units> [<count>]]]]]*

The EM float typically samples during descent and ascent phases.

For example, to sample the EM hardware every 20 seconds during Park Descent, Deep Descent and Ascent between 200 dbar and 0 dbar, enter:

```
<PARKDESCENT>
SAMPLE EM 4 300 20 SEC
SAMPLE PTS 4 302 2 DBAR
PROFILE CTD 4 305 2 1 HZ
<ASCENT>
SAMPLE EM 300 5 20 SEC
SAMPLE PTS 300 4 2 DBAR
PROFILE CTD 300 4 2 1 HZ
```

For more information on using the sample.cfg file to program the behavior of the float along with additional examples, refer to “sample.cfg File” on page 4-2.

NOTE *EM float sampling should be specified to be within the bounds of CTD continuous profiling. For example, if Park Descent Profiling is specified from 4 decibars to 500 decibars, EM sampling should be specified to start after 4 decibars and stop before 500 decibars.*

G.8 Reading the Science Log EM Data

EM data are provided in the science_log.bin file. For instructions on how to uncompress this file and convert it to a CSV file, refer to “Converting the science_log.bin and vitals_log.bin Files to CSV Files” on page 4-19.

All rows in the science_log.csv file with the first column label "EMA" contain EM data. An example row of EM data extracted from a science_log.csv file is shown in Figure G-3. However, the field names and data are displayed in a vertical format for fitting on the page. An example *partial* row of the same data is shown in Figure G-4.

Sensor type	EMA
Time stamp	20200521T175054
rotp_hx	19.92438
rotp_hy	45.09007
e1_coef40	588.87854
e1_coef41	-915.88898
e2_coef40	967.07141
e2_coef41	339.72559
e1_mean4	8714600
e2_mean4	7990270
e1_sdev4	415.66476
e2_sdev4	180.33456
buoy_pos_c0	2006.34839
hxdt_sdev	25.69936
hydt_sdev	31.0181
bt_mean	941.74878
hz_mean	827.84906
ax_mean	482.65091
ax_sdev	0.78246
ay_mean	483.53638
ay_sdev	0.68512
az_mean	235.41269
hx_mean	495.25089
hy_mean	531.73462

Figure G-3: Example Row of EM Data in a science_log.csv File, Shown Vertically

EMA	20200521T175054	19.92438	45.09007	588.87854	-915.88898	...

Figure G-4: Example *Partial* Row of EM Data in a science_log.csv File

G.9 International Geomagnetic Reference Field Model

The International Geomagnetic Reference Field (IGRF) is a series of mathematical models of the Earth's main magnetic field and its annual rate of change (secular variation) that is used to create the velocity data from the EM float science log. The IGRF is produced and updated under the direction of the International Association of Geomagnetism and Aeronomy (IAGA).

The Python pyIGRF package is used to obtain geomagnetic data from the IGRF model for any GPS location on the earth.

G.10 Processing the EM Data

The apf11_em_velocity.py script provided by Teledyne Webb Research is used to create a velocity file from the EM data in the science log. After the script runs, a velocity text file is created in the format:

```
ema-f1236a-0001-vel.txt
```

The script also has the ability to plot the data with a –p command line option.

The Python 3 packages that need to be installed to support the apf11_em_velocity.py script include:

- pyIGRF
- numpy
- csv
- datetime
- fnmatch
- getopt
- glob
- logging
- os
- sys
- time

The format of the velocity text file is the following:

```
#HDR#
# Surface GPS LAT = 28.946757
# Surface GPS LON = -88.027925
# Surface GPS Timestamp = 1493748600 # 2017-05-02 14:10:00
# Descent GPS LAT = 28.946757
# Descent GPS LON = -88.027943
# Descent GPS Timestamp = 1493748955 # 2017-05-02 14:15:55
# Descent fh = 24436.162803848278 # nT
# Descent fz = -39599.77722227217 # nT
# Descent magvar = -1.9853464674909895 # degrees
# Descent data points = 259
# Ascent GPS LAT = 28.946965
# Ascent GPS LON = -88.028678
# Ascent GPS Timestamp = 1493766895 # 2017-05-02 19:14:55
# Ascent fh = 24436.10124080716 # nT
# Ascent fz = -39599.926596396246 # nT
# Ascent magvar = -1.9849407166946007 # degrees
# Ascent data points = 342
# Constant esep1 = 0.219 # meters
# Constant esep2 = 0.219 # meters
# Constant uvpc1 = 5.960e-04 # uV / LSB
# Constant uvpc2 = 5.960e-04 # uV / LSB
# Constant alpha1 = 0.379 # radians
# Constant alpha2 = 1.95 # radians
# Constant c1e1 = 0.5
# Constant c1e2 = 0.5
# Constant c2e1 = -0.2
# Constant c2e2 = -0.2
# columns =
P,T,S,u1,v1,verr1,u2,v2,verr2,dpdt,e1mean,e2mean,buoy_pos,times
tamp
# units =
dbar,degC,PSU,m/s,m/s,m/s,m/s,m/s,m/s,m/s,uV,uV,counts,s
#DESCENT#
32.0,24.668,36.376,0.186,nan,0.003,0.219,nan,0.003,nan,-10.9,-
200.2,9.0,1493748963
34.6,24.664,36.374,0.185,0.078,0.002,0.218,0.110,0.002,0.175,-
10.3,-198.5,9.0,1493748978
37.2,24.658,36.384,0.173,0.097,0.003,0.192,0.127,0.004,0.179,-
10.2,-197.2,9.0,1493748993
```

```
40.1,24.650,36.403,0.139,0.128,0.004,0.148,0.151,0.004,0.180,-
10.0,-196.0,9.0,1493749009
#ASCENT#
1015.0,5.083,34.928,-0.064,nan,0.003,-0.067,nan,0.003,nan,4.7,-
154.1,94.0,1493756195
1013.2,5.097,34.927,-0.059,-0.002,0.001,-0.064,-0.018,0.002,-
0.127,4.7,-153.7,94.0,1493756210
1011.1,5.108,34.926,-0.051,-0.009,0.001,-0.051,-0.028,0.002,-
0.131,4.3,-153.1,94.0,1493756226
1009.1,5.120,34.926,-0.041,-0.018,0.001,-0.047,-0.034,0.001,-
0.131,4.0,-152.8,94.0,1493756241
1007.0,5.139,34.928,-0.039,-0.025,0.001,-0.041,-0.039,0.001,-
0.131,3.8,-152.5,94.0,1493756257
```

NOTE *GPS data are needed for velocity file calculations. Depending on how the float is deployed, when initially deployed,* ***PreludeTime*** *should be set to allow the float to be at the surface long enough to get the initial GPS reading.*

G.11 Reading EM Raw Data

The ema_log.bin file contains raw EM float data associated with each EM sample in the science_log.bin file. The file is transmitted in the format:

```
apf11.000.20200518T183746.ema_log.bin.gz
```

G.12 Converting the ema_log.bin File to an CSV File

The ema_log.bin file must be uncompressed and then converted to a CSV file before it can be opened in a spreadsheet program.

To convert the ema_log.bin file to a CSV file:

1. Locate the ema_log.bin.gz file in the home directory of the server. The actual file names will additionally include the float name, the profile number and a timestamp as described in “Log Files” on page 4-17 .
2. Enter `gunzip <filename>`.

   The specified file will be unzipped to the corresponding ema_log.bin file with no .gz extension.
3. Enter `python apf11dec.py <filename>`.

   The specified binary ema_log.bin will be used to generate a corresponding ema_log.csv file.
4. Open the CSV file in a spreadsheet program, such as Microsoft Excel.

G.13 Show Command

The **Show** command displays all of the information available from the EM hardware. To use the **Show** command, a PC must be connected to the float and running a terminal program.

To display the EM hardware configuration information, enter:

```
ema_show
```

The EM hardware configuration will be displayed on the PC as in the following example:

```
> ema_show
20200606T113556|5|T_CMD|ema_show
20200606T113556|5|EMA|sensor has no settings to obtain
ReqSamples:          50
SlideSamples:        25
EMALogMaxPressure:   100.00
EMALogMissionCycle:  10
EMA show PASSED
```

G.14 Other Commands

The following additional EM float commands are available:

ema_config. Runs the configuration sequence for the EM hardware.

Syntax: ema_config

ema_sample. Obtains sample data from the EM hardware.

Syntax: ema_sample

ema_test. Runs through configuration, displays configuration and obtains sample data from the EM hardware.

Syntax: ema_test

G.15 EM Hardware Factory Configuration

There are no EM hardware factory configuration parameters.

APPENDIX H: Diagnostic Messages Example

When starting the mission from a PC or running the self test from a PC, progress and diagnostic messages will be displayed on the PC as in the example below. Tests that have passed are indicated as "<<PASS>>," and tests that have failed are indicated as "<<FAIL>>."

```
> sys_self_test
20200604T200241|5|T_CMD|sys_self_test


********************************************************************************
System Self Test Started @ 2020-06-04 20:02:43
********************************************************************************


Float ID: f1234
Firmware Version: 03/05/20 14:04:09 2.14.3 STD Final
-------------------------------------Vitals-------------------------------------
Battery Voltage:     15.13 V
Float Current:       33.0490 mA
Coulomb Count:       242.9231 mAh
Bladder Pressure:    8.0 dbar
Internal Vacuum:     8.1 dbar
Buoyancy Position:   262 counts
--------------------------------------------------------------------------------

---------------------------------Mission Config---------------------------------
ActivateRecoveryMode off
AscentRate 0.08
AscentStartTimes -1 -1 -1 -1
AscentTimeout 521
AscentTimerInterval 300
BuoyancyNudge 100
DeepDescentCount 262
DeepDescentPressure 2000.00
DeepDescentTimeout 334
DeepDescentTimerInterval 300
DeepProfileFirst on
DownTime 13953
EmergencyTimerInterval 3600
HyperRetractCount 0
HyperRetractPressure 850.00
IceBreakupDays 14
IceCriticalT -1.78
IceDescentCycles 0
IceDetectionP 50.00
IceEvasionP 20.00
IceMonths 0FFD
IdleTimerInterval 3600
```

```
InitialBuoyancyNudge 300
LeakDetect on
LogVerbosity 5
MActivationCount 262
MActivationPressure 25.00
MinBuoyancyCount 147
MinVacuum 9.00
ParkBuoyancyNudge 10
ParkDeadBand 25.00
ParkDescentCount 1081
ParkDescentTimeout 334
ParkDescentTimerInterval 3600
ParkPressure 1000.00
ParkTimerInterval 3600
PnPCycleLen 1
PreludeSelfTest on
PreludeTime 120
SurfacePressure 4.00
TelemetryDays 0 0
TelemetryInterval 900
TelemetryTimeout 120
UpTime 641
VitalsMask 0007
Checksum 18CA
--------------------------------------------------------------------------------

---------------------------------Sample Config----------------------------------
<PARK>
SAMPLE PTS 2500 4 0 DBAR 1
<ASCENT>
SAMPLE PTS 2000 400 50 DBAR 1
SAMPLE PTS 400 360 20 DBAR 1
SAMPLE PTS 360 20 10 DBAR 1
SAMPLE PTS 20 5 5 DBAR 1
PROFILE CTD 2100 4 2 1 HZ #Regime:1
--------------------------------------------------------------------------------

---------------------------------System Config----------------------------------
float_id f1234
air_bladder_max 12
buoyancy_pump_min 147
buoyancy_pump_max 3705
iridium f1234 iridium 00881600000000 rudics 00881600000000 dialup
--------------------------------------------------------------------------------

---------------------------------Sensors Config---------------------------------
sensor seabird_ctd_41cp COM2 B9600
gps gps15xh
modem iridium9523
--------------------------------------------------------------------------------

                                 Report System Configuration : << PASS >>
--------------------------------------------------------------------------------
```

```
Air Engine Test:
Deflating Air Bladder prior to starting test
20200604T200309|5|AIR|deflated
Air Bladder deflated
Starting values:
        Internal Vacuum:        8.1 dbar / 11.8 PSI
        Air Bladder:            8.0 dbar / 11.7 PSI
        Battery Voltage:        15.1 V
Inflating Air Bladder:
Air Engine started
Air Engine stopped at 12.12 dbar
print_vitals_data|vitals=Air1|pulses=22|runTime=110|minBattV(V)=14.93|minBattV(cnts)=0|
avgBattV(V)=14.94|avgBattV(cnts)=3397
print_vitals_data|vitals=Air2|maxBattC(mA)=135.79|avgBattC(mA)=123.30|maxPumpC(mA)=100.
22|maxPumpC(mA)=14.94
print_vitals_data|vitals=Air3|maxPumpC(cnts)=836|avgPumpC(mA)=94.08|avgPumpC(cnts)=764
        Air Engine Pulses:      22
        Air Engine Coulombs:    247.611 mAh
        Air Engine Avg Current: 123.303 mA
        Air Engine Max Current: 135.790 mA
        Battery Avg Voltage:    14.94 V
        Battery Min Voltage:    14.93 V
        Internal Vacuum:        7.2 dbar / 10.4 PSI
        Air Bladder Pressure:   12.1 dbar / 17.5 PSI
Deflating Air Bladder at end of test
20200604T200603|5|AIR|deflated
Air Bladder deflated
                                             Air Engine Test : << PASS >>
--------------------------------------------------------------------------------

Buoyancy Engine Test:
Testing ASCENT Buoyuancy Adjustment
20200604T200609|5|BuoyEngine|adjusting from 263 to 618
Buoyancy Position: 265
Buoyancy Position: 274
Buoyancy Position: 290
Buoyancy Position: 304
Buoyancy Position: 318
Buoyancy Position: 332
Buoyancy Position: 350
Buoyancy Position: 367
Buoyancy Position: 384
Buoyancy Position: 400
Buoyancy Position: 416
Buoyancy Position: 429
Buoyancy Position: 444
Buoyancy Position: 459
Buoyancy Position: 475
Buoyancy Position: 489
Buoyancy Position: 506
Buoyancy Position: 524
Buoyancy Position: 541
Buoyancy Position: 554
Buoyancy Position: 569
```

```
Buoyancy Position: 582
Buoyancy Position: 597
Buoyancy Position: 613
20200604T200807|5|BuoyEngine|destination 621 reached after 01:57
print_vitals_data|vitals=Bouy1|runTime=121|potDelta=307|minBattV(V)=14.79|minBattV(cnts
)=0|avgBattV(V)=14.83
print_vitals_data|vitals=Bouy2|avgBattV(cnts)=3372|maxBattC(mA)=214.47|avgBattC(mA)=180
.28|maxPumpC(mA)=202.71
print_vitals_data|vitals=Bouy3|maxPumpC(cnts)=444|avgPumpC(mA)=184.17|avgPumpC(cnts)=38
9
ASCENT buoyancy adjustment vitals:
        Buoyancy Coulombs:     7.187 mAh
        Buoyancy Avg Current:  180.279 mA
        Buoyancy Max Current:  214.467 mA
        Battery Avg Voltage:   14.83 V
        Battery Min Voltage:   14.79 V
Testing DESCENT Buoyancy Adjustment
20200604T200817|5|BuoyEngine|adjusting from 621 to 266
Buoyancy Position: 620
Buoyancy Position: 609
Buoyancy Position: 594
Buoyancy Position: 579
Buoyancy Position: 565
Buoyancy Position: 550
Buoyancy Position: 535
Buoyancy Position: 519
Buoyancy Position: 502
Buoyancy Position: 485
Buoyancy Position: 470
Buoyancy Position: 455
Buoyancy Position: 442
Buoyancy Position: 428
Buoyancy Position: 414
Buoyancy Position: 398
Buoyancy Position: 380
Buoyancy Position: 364
Buoyancy Position: 347
Buoyancy Position: 331
Buoyancy Position: 316
Buoyancy Position: 301
Buoyancy Position: 287
Buoyancy Position: 274
20200604T201016|5|BuoyEngine|destination 265 reached after 01:58
print_vitals_data|vitals=Bouy1|runTime=122|potDelta=306|minBattV(V)=14.79|minBattV(cnts
)=0|avgBattV(V)=14.82
print_vitals_data|vitals=Bouy2|avgBattV(cnts)=3370|maxBattC(mA)=197.12|avgBattC(mA)=158
.82|maxPumpC(mA)=177.49
print_vitals_data|vitals=Bouy3|maxPumpC(cnts)=391|avgPumpC(mA)=159.20|avgPumpC(cnts)=34
0
DESCENT buoyancy adjustment vitals:
        Buoyancy Coulombs:     6.562 mAh
        Buoyancy Avg Current:  158.816 mA
        Buoyancy Max Current:  197.122 mA
        Battery Avg Voltage:   14.82 V
        Battery Min Voltage:   14.79 V
```

```
                                        Buoyancy Engine Test : << PASS >>
--------------------------------------------------------------------------------

GPS Test:
Modem attempting to register with network
Modem registered after 2 attempts
20200604T201047|5|COMMS|found sky
20200604T201049|5|GPS|Updating Almanac for 15 minutes
20200604T202550|5|GPS|Completed Almanac Update on 06/04/2020 20:25:50
20200604T202550|5|GPS|Next Almanac Update 09/02/2020 20:25:50
20200604T202601|5|RMC|Set Clock: 06/04/2020 20:26:01
20200604T202601|5|GPS|GPS TimeToFix: 4 secs
20200604T202601|5|GPS|GPS Skew: 1 secs
20200604T202601|5|GPS|GPS Fix: 06/04/2020 20:26:01,41.64016,-70.61053,10
GPS|Lat:41.64016|Lon:-70.61053|NSats:10
20200604T202604|5|GPS|Time and location set
GPS time and location updated
RF Board Max Current: 31.3 mA
Battery Min Voltage:  15.0 V
                                                    GPS Test : << PASS >>
--------------------------------------------------------------------------------

Battery Voltage @ 15.10 V (above 11 V) : PASS
Internal Vacuum @ 8.24 dbar (below 9.00 dbar) : PASS
Leak Detection Voltage @ 2.35 V (above 2 V) : PASS
Relative Humidity @ 5.15 percent (below 75 percent) : PASS
                                                 Vitals Test : << PASS >>
--------------------------------------------------------------------------------

SBE CTD Test:
tswait=20
outputpts=1
includetransitionbin=0
includenbin=1
autobinavg=0
top_bin_interval=2
top_bin_size=2
top_bin_max=10
middle_bin_interval=2
middle_bin_size=2
middle_bin_max=20
bottom_bin_interval=2
bottom_bin_size=2
pcutoff=2.00
                                           SBE CTD Configure : PASS
Model:                     SBE 41CP
Version:                   7.2.5
Serial Number:             11741
Variant:                   CP UW
tswait:                    20
pcutoff:                   2.0
automatic bin averaging:   no
nsamples:                  18
```

```
nbins:                    0
top bin interval:         2
top bin size:             2
top bin max:              10
middle bin interval:      2
middle bin size:          2
middle bin max:           20
bottom bin interval:      2
bottom bin size:          2
include transition bin:   no
include samples per bin:  yes
real-time output:         PTS
temperature:  09-May-19
        TA0:                -8.600550e-04
        TA1:                2.953019e-04
        TA2:                -3.817722e-06
        TA3:                1.541883e-07
conductivity:  09-May-19
        G:                  -1.007748e+00
        H:                  1.229747e-01
        I:                  -9.665833e-05
        J:                  2.219393e-05
        CPCOR:              -9.570000e-08
        CTCOR:              3.250000e-06
        WBOTC:              -5.818992e-08
pressure S/N = 11129406, range = 2900 psia:  05-Apr-19
        PA0:                3.009636e-01
        PA1:                3.894178e-04
        PA2:                -2.862779e-13
        PTCA0:              -1.340698e+04
        PTCA1:              5.648351e+01
        PTCA2:              -9.159981e-01
        PTCB0:              3.122886e+05
        PTCB1:              1.826021e+01
        PTCB2:              -1.435943e-01
        PTHA0:              3.125957e+02
        PTHA1:              -6.296966e-05
        PTHA2:              -1.047200e-12
        POFFSET:            0.000000e+00
                                           SBE CTD Get Info : PASS
                                      SBE CTD Verify Config : PASS
Pressure:-0.23
pressure: -0.23 dbar
Pressure:-0.24|Temperature:25.848
pressure: -0.24 dbar, temperature: 25.8484 degC
20200604T202631|4|CTD|PTS sample not taken, pressure at surface. -0.23 < 4.00
                                             SBE CTD Sample : PASS
                                               SBE CTD Test : << PASS >>
--------------------------------------------------------------------------------

Modem Test:
Modem               : Iridium
Model               : 9523N
```

```
Serial Number       : 300000000000000
SIM ICCID           : 898800000000000
Call Processor Version: DB17011
Modem DSP Version: 1.7 svn: 2358
Audio DSP Version: 1.7 svn: 2459
DBB Version: 0x0001 (ASIC)
Main RFA Version: 0x0007 (SRFA2)
Aux RFA Version: 0x07ff (Not present)
NVM Version: KVS
BOOT Version: BOOT0fa3/9523NRevE/02/RAW15
                                              Get Modem Info : PASS
Testing Primary Number: 008816000000000 (RUDICS)
20200604T202716|5|COMMS|attempt 1/5 to upload \temp\test.20200604T202653.txt
Modem attempting to register with network
Modem registered after 2 attempts
20200604T202733|5|COMMS|CSQ=5
Attempting to connect to 00881600000000 (RUDICS)
Connected
uploaded test.20200604T202653.txt
uploaded 67 bytes in 13 secs at 5.1538 bytes/sec
20200604T202815|4|COMMS|logout unsuccessful, connection lost
                                                     Connect : PASS
                                                       Login : PASS
                                                 File Upload : PASS
                                                   Send File : PASS
20200604T202841|5|COMMS|attempt 1/5 to download test.20200604T202653.txt
Modem attempting to register with network
Modem registered after 2 attempts
20200604T202857|5|COMMS|CSQ=5
Attempting to connect to 00881600000000 (RUDICS)
Connected
downloading test.20200604T202653.txt
downloaded test.20200604T202653.txt
downloaded 128 bytes in 3 secs at 42.6667 bytes/sec
                                                     Connect : PASS
                                                       Login : PASS
                                               File Download : PASS
                                                Receive File : PASS
                                           File Compare Size : PASS
                                       File Compare Contents : PASS
                                         Primary Number Test : PASS
================================================================================

Testing Secondary Number: 00881600000000 (DIALUP)
20200604T203026|5|COMMS|attempt 1/5 to upload \temp\test.20200604T203003.txt
Modem attempting to register with network
Modem registered after 2 attempts
20200604T203045|5|COMMS|CSQ=5
Attempting to connect to 00881600000000 (DIALUP)
Connected
uploaded test.20200604T203003.txt
uploaded 67 bytes in 12 secs at 5.5833 bytes/sec
20200604T203144|4|COMMS|logout unsuccessful, connection lost
```

```
                                                 Connect : PASS
                                                   Login : PASS
                                             File Upload : PASS
                                               Send File : PASS
20200604T203210|5|COMMS|attempt 1/5 to download test.20200604T203003.txt
Modem attempting to register with network
Modem registered after 2 attempts
20200604T203227|5|COMMS|CSQ=4
Attempting to connect to 00881600000000 (DIALUP)
Connected
20200604T203342|4|COMMS|login attempt 1/10 failed, username prompt not received
downloading test.20200604T203003.txt
downloaded test.20200604T203003.txt
downloaded 128 bytes in 8 secs at 16.0000 bytes/sec
20200604T203449|4|COMMS|logout unsuccessful, connection lost
                                                 Connect : PASS
                                                   Login : PASS
                                           File Download : PASS
                                            Receive File : PASS
                                       File Compare Size : PASS
                                   File Compare Contents : PASS
                                   Secondary Number Test : PASS
                       Primary and Secondary Number Test : PASS
                                              Modem Test : << PASS >>
--------------------------------------------------------------------------------

-------------------------------------Vitals-------------------------------------
Battery Voltage:     14.99 V
Float Current:       37.2680 mA
Coulomb Count:       298.5481 mAh
Bladder Pressure:    8.2 dbar
Internal Vacuum:     8.3 dbar
Buoyancy Position:   264 counts
--------------------------------------------------------------------------------

********************************************************************************
System Self Test PASSED @ 2020-06-04 20:35:02
```